



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorthesis

Maximilian Wichern

Einsatz von kabellosen MESH Netzwerken in
der smarten Beleuchtungssteuerung an Bord
von Schiffen

Maximilian Wichern

Einsatz von kabellosen MESH Netzwerken in der
smarten Beleuchtungssteuerung an Bord von
Schiffen

Bachelorthesis eingereicht im Rahmen der Bachelorprüfung
im Studiengang Informations- und Elektrotechnik
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Björn Ingo Lange
Zweitgutachter : Dipl.-Ing. Patrick Nösler

Abgegeben am 8. August 2018

Maximilian Wichern

Thema der Bachelorthesis

Einsatz von kabellosen MESH Netzwerken in der smarten Beleuchtungssteuerung an Bord von Schiffen

Stichworte

Beleuchtung, kabellose Steuerung, Smart, Industrie 4.0, ZigBee, MESH Netzwerk, XBee, IoT, Sensoren

Kurzzusammenfassung

Diese Arbeit umfasst die ZigBee-Protokoll basierte smarte Beleuchtungssteuerung an Bord von Schiffen, in der es darum geht die Beleuchtung auf Schiffen nicht nur durch Kabeleinsparung, sondern auch durch intelligente Verwendung von Sensoren und Aktoren individueller zu gestalten und anzupassen. Mit dieser Arbeit wird die technische Machbarkeit an Bord von Schiffen nachgewiesen und der Nutzen in der Industrie dargestellt.

Maximilian Wichern

Title of the paper

Usage of wireless MESH networks in smart lighting control on board vessel

Keywords

Lighting, wireless control, smart, Industrie 4.0, ZigBee, MESH Network, XBee, IoT, sensors

Abstract

Inside this report the wireless and smart lighting control with the ZigBee protocol on board vessel is described. Beside the economization of cables you get the intelligent usage of sensor and actuators to be more individual. With this report the technical viability on board the vessel and the industrial usage will be shown.

Abkürzungsverzeichnis

Abk. Abkürzung

ADC Analog-Digital-Converter

DALI Digital Addressable Lighting Interface

EVG elektronisches Vorschaltgerät

GUI Graphical User Interface

I2C Inter-Integrated Circuit

IC Integrated Circuit

I/O Input / Output

IoT Internet of Things

KVG konventionelles Vorschaltgerät

LLL Low Location Light

OP Operationsverstärker

OSI Open Systems Interconnection

PCB Printed Circuit Board

PWM Puls-Weiten-Modulation

Ro-Ro Roll on Roll off, Transportschiffe für Pkw und Lkw

SPI Serial Peripheral Interface

SPS Speicherprogrammierbare Steuerung

UART Universal Asynchronus Receiver Transmitter

VG Vorschaltgerät

z.B. zum Beispiel

Inhaltsverzeichnis

Tabellenverzeichnis	7
Abbildungsverzeichnis	8
1. Einführung	10
1.1. Zielsetzung der Thesis	12
1.2. Dokumentenaufbau	12
2. Theoretische Grundlagen	14
2.1. Physikalische Betrachtung von Licht	14
2.2. Beschreibung von Beleuchtungsanlagen auf Schiffen	16
2.3. Heute verwendete Leuchtmittel	17
2.4. Funktionalität eines Vorschaltgerätes	18
2.5. Dimmen von Leuchtmitteln	20
2.5.1. Dimmschnittstellen	20
2.5.2. Dimmverfahren	21
2.6. Mikrocontroller	24
2.7. Funknetze	25
2.7.1. Funkmodule	25
2.7.2. Wireless Local Area Network (WLAN)	26
2.7.3. ZigBee-Protokollstandard	26
2.7.4. MESH-Netzwerk	27
2.8. Theoretische Darstellung von grundlegenden Schaltungs- und Verfahrensvarianten	29
2.8.1. Super-Kondensatoren	29
2.8.2. Passive Filter	31
2.8.3. Operationsverstärker	33
3. Anforderungen und Konzepte	39
3.1. Anforderungen an das System	39
3.2. Konzept für die Schalt- und Versorgungsplatine der Leuchtenmodule	40
3.2.1. Konzept für die Spannungsversorgung der Platine	40
3.2.2. Konzept für das Schalten der Leuchten	40
3.2.3. Konzept für das Dimmen der Leuchten	40

3.2.4. Konzept für den Anschluss von Sensoren an der Platine	40
3.3. Konzept für die Schalt- und Versorgungsplatine der Tastermodule	41
3.3.1. Konzept für die Spannungsversorgung der Platine	41
3.3.2. Konzept für die Steuerungsbefehle an die Leuchtenmodule	41
3.4. Konzept für die Web-Visualisierung	41
4. Realisierung der Konzepte	42
4.1. Definition der Datenübertragung und Datenverarbeitung	42
4.1.1. Beschreibung vom XBee S2C Funkmodul	42
4.1.2. Beschreibung vom Atmel Mega328P	44
4.2. Entwicklung von Printed Circuit Boards	46
4.3. Schalt- und Versorgungsplatine der Leuchten	46
4.3.1. Beschreibung der verwendeten Schaltungen	47
4.4. Schalt- und Versorgungsplatine der Taster	55
4.4.1. Schaltung zum Senden des Schaltbefehls	55
4.4.2. Schaltung für die Batterieüberwachung	56
4.5. Liste der verwendeten XBee-Module	59
4.6. Visualisierung und Informationsweiterleitung	59
4.7. Softwarerealisierung	60
5. Auswertung	64
5.1. Technische Funktionalität	64
5.2. Funktionaler Abschlusstest des gesamten Systems	64
5.2.1. Funktionalität der Schaltfunktion	64
5.2.2. Funktionalität der Dimmfunktion	65
5.2.3. Funktionalität der Sensorik	65
5.2.4. Funktionalität des Notbetriebes	66
5.2.5. Funktionalität der Batterieüberwachung	66
5.3. Funktionalität der Visualisierung	66
6. Fazit	68
7. Ausblick	69
Literaturverzeichnis	71
A. Inhalt der CD	73
B. Quellcode	74
B.1. Quellcode Leuchtenmodul	74
B.2. Quellcode SPS-Modul	82

Tabellenverzeichnis

2.1. Vergleichstabelle der Eigenschaften unterschiedlicher Leuchtmittel	18
4.1. Messreihe zur Kalibrierung des Lichtstärkesensors mit dem Luxmeter <i>testo</i> 540	52
4.2. Übersichtstabelle über verwendete XBee-Module	59
5.1. Tabelle der technischen Funktionalität	64
A.1. Tabelle der technischen Funktionalität	73

Abbildungsverzeichnis

2.1. Elektromagnetisches Spektrum	14
2.2. Vergleich der im Licht von Halogen-, LED- und Energiesparleuchten enthaltenen Lichtspektren im Vergleich zum Lichtspektrum von Tageslicht . .	15
2.3. Schaltung eines Pulsweitenmodulators	21
2.4. Funktionsweise des PWMs	21
2.5. Darstellung der Schaltsymbole eines Triacs (rechts) und eines Diacs (links) .	22
2.6. Einfache Phasenanschnittsteuerung	23
2.7. Beispielhafter Mikrocontroller Atmel Mega328P	24
2.8. ZigBee Protokollstack	27
2.9. Beispiel einer MESH-Map	28
2.10. Bewertungstabelle der Verbindungen der Software XCTU	29
2.11. Schematische Darstellung vom Laden und Entladen eines Doppelschicht-Super-Kondensators	30
2.12. RC-Tiefpass Schaltung	31
2.13. Bodeplot eines RC-Tiefpass	32
2.14. Schaltsymbole und Übertragungsgleichungen der vier Operationsverstärker	34
2.15. Schaltung eines nicht invertierenden Operationsverstärkers	35
2.16. Schaltung eines Transimpedanz-Verstärkers	36
2.17. Schaltsymbol eines Komparators	38
4.1. XBee SC2 Funkmodul der Firma DIGI	42
4.2. Pinbelegung des XBee SC2 Funkmoduls	43
4.3. Arduino Pro Mini 3,3V/8MHz	45
4.4. Gesamte Platine der Leuchtenmodule	46
4.5. Schaltplan zum Schalten der Leuchtmittel	47
4.6. Operationsverstärkerschaltung zur Dimmung eines beliebigen 1-10V EVGs über ein PWM-Signal	48
4.7. Schaltplan der angebotenen Sensorik	50
4.8. Schematische Darstellung des Datenprotokolls des digitalen Sensors	50
4.9. Kalibrierung des analogen Lichtstärkesensors (Markierung 2, rechtes Bild) mit dem Luxmeter <i>testo 540</i> (Markierung 1, rechtes Bild)	51
4.10. spektrale Empfindlichkeit der BPW 34 Fotodiode	53
4.11. Schaltplan für den Notbetrieb	53
4.12. Zeitmessung der Versorgung durch die Kondensatoren	54

4.13. Gesamte Platine der Tastermodule	55
4.14. Schaltplan zum Senden eines Schalt- oder Dimmbefehls	55
4.15. Schaltplan der Batterieüberwachung	56
4.16. Verbrauchsmessung des Tastermoduls	57
4.17. Ansicht der Web-Visualisierung	60
4.18. Teil 1 des Programmablaufplans für die Verarbeitung der erhaltenen Schalt- befehle	61
4.19. Teil 2 des Programmablaufplans für die Verarbeitung der erhaltenen Schalt- befehle	62
4.20. Programmablaufplan für die Konvertierung von UART auf Modbus-RTU . .	63

1. Einführung

Weltweit gibt es Anfang 2016 rund 51.000 fahrtüchtige Schiffe in der Welthandelsflotte mit einer Bruttoreaumzahl¹ von über 300 BRZ, welche sich hauptsächlich in die folgenden Schiffstypen aufteilen [7]:

- Tanker
- Containerschiffe
- Roll on Roll off, Transportschiffe für Pkw und Lkw (Ro-Ro)
- Passagierschiffe
- Stückgutschiffe
- Massengutschiffe

Des Weiteren gibt es etliche private Megayachten, Arbeitsschiffe wie z.B. Bagger, Kabel- und Rohrleger. Schiffe nationaler Vollzugsbehörden und der Landesverteidigung (Zoll, Küstenschutz Marine, etc.) bilden einen weiteren eigenständigen Bereich.

Auf jedem Schiff ist heutzutage eine elektrische Beleuchtungsanlage vorhanden, welche bislang über Versorgungs- und Steuerungskabel miteinander und mit der Haupt- und Notversorgungsverteilung verbunden wird.

Mit der zunehmenden Digitalisierung und der Entwicklung des Internet of Things (IoT) findet gerade in der Haus- und Gebäudetechnik ein starker Wandel der bislang einfach gehaltenen Beleuchtung statt. Es geht dabei vermehrt um die intelligente Verwendung der gut verteilten Leuchtkörper im Raum. Durch mögliche Nutzung von Sensordaten wie z.B. Bewegungsmelder und Temperatursensoren etc., lassen sich viele Informationen gewinnen, die auch Drittsystemen zur Verfügung gestellt werden können. Hierfür sind der Anwendung kaum Grenzen gesetzt. Der Umgang mit den großen Datenmengen und die Kompatibilität der unzähligen Lösungen untereinander stellen viele Hersteller und Marktteilnehmer vor Probleme. Auf dem Markt gibt es unzählige Soft- und Hardwarelösungen die sich dessen angenommen haben.

Große Konzerne wie Phillips, Osram, „Wir Sind Heller“ und EnOcean beschäftigen sich ihrerseits mit produktreifen Lösungen der kabellosen Beleuchtungssteuerung über

¹BRZ - ist eine dimensionslose Zahl für das Raummaß eines Schiffes

Funkprotokolle und der intelligenten Anbindung von Sensoren. Beispielsweise Bewegungssensoren oder auch die Anpassung der Lichtfarbe je nach Tageszeit bringen neue Energieeinsparungsmöglichkeiten bzw. eine Steigerung der Arbeitseffizienz von Mitarbeitern, wie Henri Juslén in seiner Dissertation [9] aus 2007 belegt. Zudem können Lichtstärkesensoren genutzt werden, um die Alterung von Leuchtmittel zu überprüfen.

Wärtsilä SAM Electronics GmbH (WSAM) mit Hauptsitz in Hamburg zählt mit rund 600 Mitarbeitern zu den Größten 200 Hamburger Unternehmen [12]. Mit dem Schiffbau als einzigem Geschäftsfeld arbeitet WSAM als Dienstleister und wird von Werften auf der ganzen Welt beauftragt, das Elektrotechnik-Engineering ihrer Projekte abzuwickeln. Neben den Produktlinien arbeitet WSAM als TurnKey-Anbieter und System-Integrator. Eine TurnKey-Lösung bedeutet die komplette elektrische Ausrüstung vom Schiffsantrieb bis hin zur Kabelverlegung und dem Anschluss von Leuchten und Steckdosen mit Hilfe von Subunternehmern. Durch dauerhafte Baustellenbüros auf deutschen Werften, wie der Flensburger Schiffbau Gesellschaft, ist WSAM besonders deutschlandweit einer der Marktführer.

Diese Arbeit beschäftigt sich mit der Umsetzung von kabelloser und smarter Beleuchtungssteuerung über MESH-Netzwerke an Bord von Schiffen. Dabei werden die technische Machbarkeit und zukünftige Realisierungsmöglichkeiten aufgezeigt. Hierzu wird das öffentliche ZigBee-Protokoll (siehe Kapitel 2.7.3) verwendet, um eine bidirektionale Kommunikation zwischen den Geräten herzustellen. Die technische Machbarkeit wird in der Arbeit „ZigBee-Based Sensor Network for Shipboard Environments“ von H. Kdouh, G. Zaharia, C. Brousseau und G. El Zein [10] aus dem Jahr 2011 nachgewiesen. Sie weist darüber hinaus eine verträgliche Koexistenz von ZigBee und WiFi an Bord von Schiffen nach. Zudem schrieben Sie, dass die Signalverluste der kabellosen Kommunikation beherrschbar sind und es kaum Probleme beim Durchdringen von Stahlschotts gibt. Auf die weitreichenden Möglichkeiten von MESH-Netzwerken wird in der Arbeit aber nicht weiter eingegangen.

Eine erste Patentrecherche ergab keine wesentlichen Einschränkungen für dieses Konzept. Die Recherche wurde beim Deutschen Patent- und Markenamt sowie beim Europäischen Patentamt durchgeführt. Dabei wurden keine eindeutigen Patente im Hinblick auf eine ZigBee-MESH-Netzwerk basierende Beleuchtungssteuerung gefunden.

Ein konkretes Beispiel [4] zeigt lediglich die technische Lösungen einer Beleuchtungssteuerung die wahlweise kabelgebunden oder kabellos kommunizierend realisiert werden kann (siehe Anhang A).

1.1. Zielsetzung der Thesis

Ziel dieser Arbeit ist es, den oben genannten Wandel in der Haus- und Gebäudetechnik auf die Beleuchtungsanlage an Bord von Schiffen zu übertragen und die technische Machbarkeit unter Marineanforderungen zu zeigen. Hierzu sollen zwei prototypische Platinen entwickelt und getestet werden. Das über Funk kommunizierende System soll die Anforderung des Schaltens und Dimmens von Leuchtmitteln erfüllen, eine Anbindung von Sensorik ermöglichen und die in dieser Arbeit verwendeten Leuchtmittel in einen Sicherheitszustand bringen können. Die Notbeleuchtung ist eine Forderung der Klasse und muss an Bord von Schiffen umgesetzt werden. Hierzu sollen die Leuchtmittel im Notstromkreis bei Detektion eines Spannungsabfalls z.B. durch den Ausfall einer Hauptstromverteilung, auf 100% eingestellt werden, um den Notbetrieb sicherzustellen. Die verschiedenen Zustände der Endgeräte und Messwerte durch die Sensoren sollen für das Automationssystem des Schiffes zur Verfügung gestellt werden sowie über einen Webserver visualisiert werden. Zudem soll ein Ausblick geschaffen werden der weiterführend zeigt, welche Möglichkeiten diese Arbeit im Hinblick auf Energieeinsparung, Wirtschaftlichkeit, Prozessoptimierung und intelligente Nutzung der Beleuchtung bietet.

1.2. Dokumentenaufbau

Zunächst werden die theoretischen Grundlagen im Kapitel 2 beschrieben, die in dieser Arbeit eine Anwendung finden. Darin werden neben der physikalischen Beschreibung von Licht, der allgemeinen Beschreibung von Beleuchtungsanlagen auf Schiffen und der Beschreibung von Funknetzen auch grundlegende Schaltungs- und Verfahrensvarianten wie Super-Kondensatoren und Operationsverstärker erläutert.

Im Kapitel 3 werden nach der Aufstellung der Anforderungen an das System, die einzelnen Konzepte zu den Teillösungen wie z.B. dem Konzept zum Schalten und der Spannungsversorgung beschrieben. Dort wird erfasst, welche Anforderungen die Realisierung erfüllen muss, um das jeweilige Problem zu lösen.

Darauf folgt die Realisierung der Konzepte in Kapitel 4. Hier wird die technische Umsetzung festgelegt und genauer beschrieben, wie und mit welchen Bauteilen das jeweilige Problem gelöst wurde.

Die Auswertung in Kapitel 5 fasst die Ergebnisse anhand eines Testplans zusammen und beschreibt den funktionalen Abschlusstest des gesamten Systems.

Ein abschließendes Fazit in Kapitel 6 bewertet die technische Umsetzung dieser Arbeit und zeigt die Probleme, die während der Bearbeitung aufgetreten sind. Hier wird zusammengefasst welche Funktionen umgesetzt werden konnten oder mehr als erfüllt bzw. nicht erfüllt wurden.

Der abschließende Ausblick in Kapitel 7 soll beispielhaft zeigen, welche weiterreichenden Entwicklungs- und Anwendungsmöglichkeiten diese Arbeit bieten kann.

2. Theoretische Grundlagen

2.1. Physikalische Betrachtung von Licht

Licht ist eine Form elektromagnetischer Strahlung und beinhaltet den Teil an Wellenlängen der für das menschliche Auge sichtbar ist. Im Allgemeinen spricht man vom elektromagnetischen Spektrum, welches sich in spezifische Wellenlängenbereiche von Strahlung unterteilt und in Abbildung 2.1¹ dargestellt ist. Die Wellenlängen sind in der Einheit Nanometer [nm] angegeben.

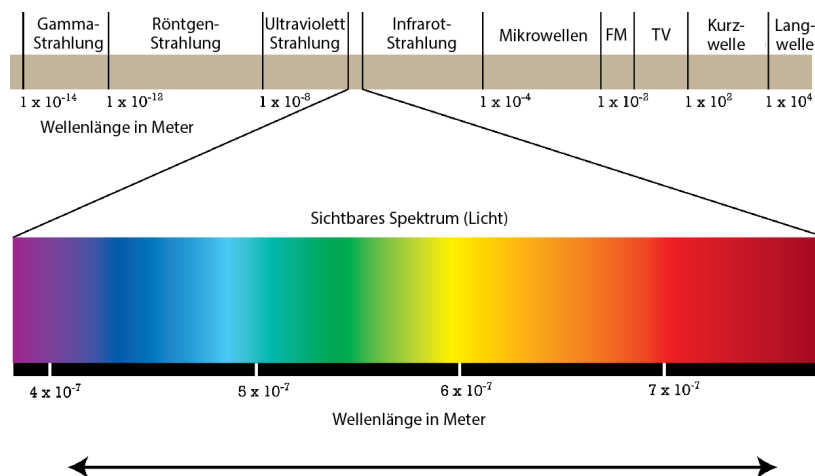


Abbildung 2.1.: Elektromagnetisches Spektrum

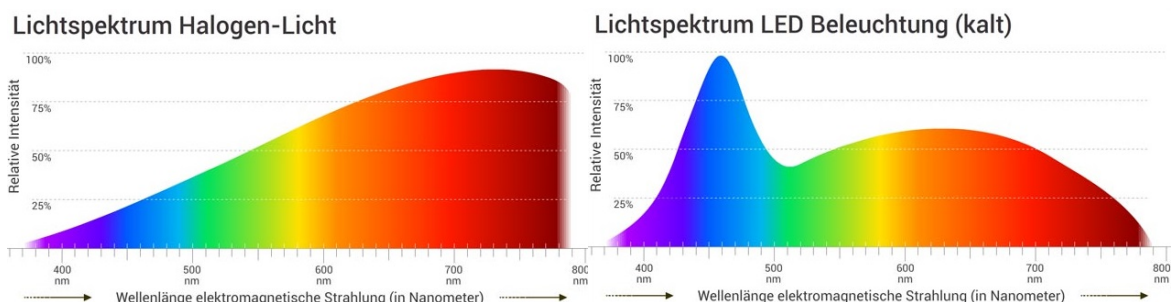
Das Spektrum reicht von der Gamma-Strahlung bis hin zur Langwelle, die von der Radiotechnologie genutzt wird.

Licht breitet sich somit als Welle aus und erreicht eine Geschwindigkeit von $c = 299792458 \frac{m}{s}$ (Lichtgeschwindigkeit) im Vakuum. Die Wellengleichung kann aus den Maxwell-Gleichungen hergeleitet werden, worauf in dieser Arbeit nicht näher eingegangen wird.

¹<http://www.thermoglass.de/de/artikel/elektromagnetisches-spektrum>

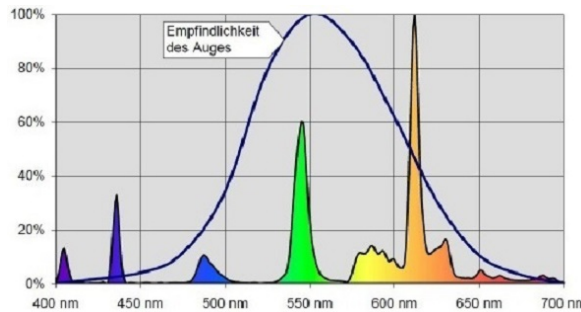
Speziell das sichtbare Licht hat eine Wellenlänge von 380 bis 780 nm. Die unterschiedlichen Rezeptortypen im menschlichen Auge, die für bestimmte Bereiche der jeweiligen Wellenlängen empfindlich sind, leiten ihr Erregungsmuster an das Gehirn weiter, wo die Informationen verarbeitet und somit Farben wahrgenommen werden.

Dabei beinhaltet das Sonnenlicht das komplette sichtbare Lichtspektrum und wird aus der Sonnenenergie erzeugt. Das in dieser Arbeit künstlich erzeugte Licht hat aufgrund der unterschiedlichen Strahlungsverhalten der möglichen Lichtquellen andere Lichtspektrum-Verteilungen wie die Abbildungen in 2.2 [1] zeigen.



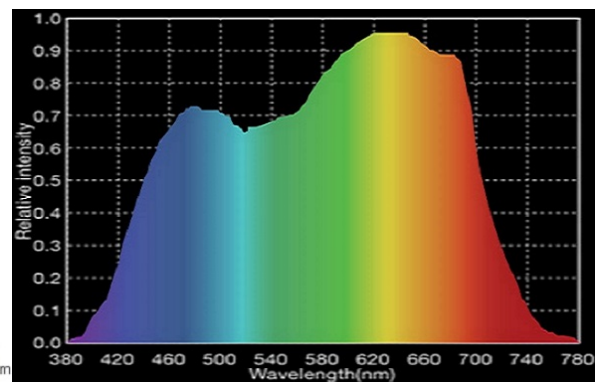
(a) Lichtspektrum von Halogen-Licht

(b) Lichtspektrum von kaltem LED-Licht



(c) Lichtspektrum einer Energiesparleuchte

Quelle: <http://www.hbernstaedt.de>



(d) Lichtspektrum von Tageslicht

Quelle: <http://www.exciteled.ch/de/spektrum-photosynthese>

Abbildung 2.2.: Vergleich der im Licht von Halogen-, LED- und Energiesparleuchten enthaltenen Lichtspektren im Vergleich zum Lichtspektrum von Tageslicht

Vergleicht man die Diagramme miteinander, so lässt sich erkennen, dass sich die Verteilungen vom LED-Spektrum und dem Tageslichtspektrum am ähnlichsten sind. Das Spektrum von Halogen-Licht weicht im Gegensatz dazu ab. Noch stärker abweichend ist das Lichtspektrum einer Energiesparleuchte. Es beinhaltet lediglich einzelne Spektralfarben blau (440 nm), grün (545 nm) und orange/rot (620 nm). Je mehr die Farbwiedergabe

der Leuchtmittel dem Farbspektrum vom Tageslicht gleicht, desto qualitativer hochwertiger bewertet das menschliche Auge die Farben.

Zudem werden die einzelnen Leuchtmittel in ihrer Farbtemperatur unterschieden. Die Farbtemperatur ist dabei ein Maß, um den Farbeindruck einer Lichtquelle zu bestimmen. Die Temperatur wird in *Kelvin* angegeben und beträgt bei einer „kalten“ LED um die 4000K. Das Sonnenlicht hat im Vergleich eine Farbtemperatur von ca. 6000K. Wenn man von einer „warmen“ Lichtquelle spricht, wird der Farbton zunehmend rötlicher und die Farbtemperatur sinkt.

2.2. Beschreibung von Beleuchtungsanlagen auf Schiffen

Seit Ende des 19. Jahrhunderts existiert die elektrische Energieversorgung auf Schiffen und mit ihr auch die elektrische Beleuchtung. Diese setzt sich heute in der Regel neben der Innenraum- und Außenbeleuchtung, aus der Navigationsbeleuchtung (Positionslichter), den Low Location Lights (LLLs), den Suchscheinwerfern sowie jeglicher Spezialbeleuchtung wie z.B. der Helikopterdeckbeleuchtung. Die LLL stellen die Sicherheits- und Fluchtwegbeleuchtung im Innenraum des Schiffes dar.

Bis heute wird die Beleuchtungsanlage überwiegend im konservativen Stil verdrahtet. Es gibt also eine kabelgebundene Verbindung zwischen der Stromverteilungsstation, dem klassischen Lichtschalter oder Dimmer und dem Leuchtkörper. Durch Speicherprogrammierbare Steuerung (SPS) werden heutzutage unter anderem relevante Informationen über die Beleuchtung wie z.B. Schaltzustände, aber auch Sensordaten gesammelt und die Steuerung von Pumpen, Lüftungssysteme usw. umgesetzt und diese visualisiert. Dabei werden gerade in der Gebäudeautomation, aber auch an Bord von Schiffen Informations- und Steuerbusse wie Digital Addressable Lighting Interface (DALI) oder KNX neben den anderen Steuerungsschnittstellen wie 0-10V, 1-10V oder der klassischen Installation verwendet. Im Kapitel 2.5 wird auf dieses Thema näher eingegangen.

Neben der Spannungsversorgung der Leuchtmittel werden dafür zweiadrige Steuerleitungen verwendet, die üblicherweise in einem $5 \times 1,5 \text{ mm}^2$ Kabel zusammengefasst sind. Die Beleuchtung innerhalb eines Raumes wird in unterschiedliche Kreise aufgeteilt. Dabei verläuft je ein Versorgungskabel von einer Hauptstrom- bzw. Notstromverteilung zur ersten Leuchte eines Beleuchtungskreises. Von dort aus werden alle Leuchten eines Kreises durchgeschleift. Ein weiteres Kabel wird zu jedem Lichtschalter und zu jeder Steckdose verlegt. In der Regel befindet sich immer ein Not- und ein Hauptbeleuchtungskreis im Raum. Bei sicherheitsrelevanten und wichtigen technischen Räumen werden in Bezug auf Redundanz auch mehrere Beleuchtungskreise vorgesehen.

Die Verteilung und ungefähre Position der vorgesehenen Leuchtkörper, Taster und Steckdosen werden in einem Beleuchtungsplan festgehalten. Im Anhang A wird beispielhaft der Beleuchtungsplan einer Ebene von einer gebauten Offshore-Umspannplattform gezeigt. Dieser dient unter anderem zusammen mit weiteren Ausführungsvorgaben als grober Montageplan für Subunternehmer. Des Weiteren wird zu jedem Projekt ein Beleuchtungskatalog erstellt, der alle verwendeten Leuchten und Steckdosen sowie Taster beinhaltet. In ihm sind die technischen Eigenschaften der Geräte festgehalten.

Nach Erstellung der dokumentationspflichtigen Unterlagen übernimmt in der Regel eine Baustelle auf der Werft die Montageleitung der Beleuchtung.

2.3. Heute verwendete Leuchtmittel

Seit Verwendung der elektrischen Beleuchtung auf Schiffen wurde die Glühlampe oder auch die Metaldampflampe in diversen Ausführungen an Bord von Schiffen verbaut. Technologisch wurde sie dann im Laufe der Zeit durch die Halogenbeleuchtung ersetzt, welche im Grunde genommen wie eine Glühfadenlampe funktioniert. Mit dem verwendeten Halogengas werden jedoch höhere Betriebstemperaturen erreicht und somit der Wirkungsgrad gesteigert.

Am 18. März 2009 legte das Europäische Parlament und Rat mit der Verordnung Nr. 244/2009 die Mindestanforderungen in Bezug auf die Energieeffizienz und die Qualität für verwendete Leuchtmittel fest². Das Ziel war die komplette Ersetzung der handelsüblichen Glühlampe durch energiesparendere Leuchtmittel. Dazu zählen Leuchtdioden, Energiesparlampen und Halogenlampen. Durch die immer größer werdende Nachfrage nach LED-Beleuchtung und dem immer mehr sinkenden Preis dafür ist es heutzutage üblich, die Beleuchtungsanlagen auf Schiffen überwiegend in LED-Technik zu realisieren. Der große Vorteil ist dabei die geringe Leistungsaufnahme.

In der folgenden Vergleichstabelle 2.1 sind die Eigenschaften der unterschiedlichen Leuchtmittel zusammengefasst und gegenübergestellt, die einen Lichtstrom wie eine 60W Glühlampe aufweisen. Die Werte wurden aus dem Wikipediaeintrag zum Thema Leuchtmittel [2] und dem E-Book von Conrad [8] entnommen.

²Quelle:https://www.umweltbundesamt.de/sites/default/files/medien/376/dokumente/verordnung_244_2009_eg_haushaltslampen_mit_ungebuendeltem_licht_0.pdf

Tabelle 2.1.: Vergleichstabelle der Eigenschaften unterschiedlicher Leuchtmittel

Leuchtmittel	Glühlampe	Halogenlampe	Energiesparlampe	Leuchtdiode
Leistungsaufnahme in Watt	40-60W	7-60W	5-15W	3-20W
Lichtausbeute in Lumen pro Watt	etwa $12 \frac{lm}{W}$	$15 - 27 \frac{lm}{W}$	$40 - 65 \frac{lm}{W}$	$80 - 150 \frac{lm}{W}$
Energieeinsparung ggü. einer Glühlampe	-	bis zu 30%	bis zu 80%	über 90%
Lebensdauer	1000 Std.	2000 bis 4000 Std.	10000 bis 15000 Std.	20000 bis 50000 Std.
dimmbar	ja	ja	ja	ja

Anhand dieser Tabelle wird noch einmal deutlich, dass die Leuchtdiode die konkurrierenden Techniken auf dem Markt mit ihrer Langlebigkeit und ihrer geringe Leistungsaufnahme immer mehr verdrängt.

Mit der Zeit führt die Alterung der Leuchtmittel dazu, dass die Leuchtstärke abnimmt. Neben der Alterung der Vorschaltgeräte und deren Bauteile kommt es durch Temperaturschwankungen und Verschmutzungen zu Verringerungen der Lebensdauer. Mit dem sogenannten Planungsfaktor wird versucht die Alterung im Vorwege einzurechnen, um die Vorgaben zur Lebensdauer zu erfüllen.

2.4. Funktionalität eines Vorschaltgerätes

Ein Vorschaltgerät wird dazu verwendet, um Leuchtmittel wie Energiesparlampen oder Leuchtdioden, die nicht direkt über das Stromnetz angebunden werden können, zu betreiben. Dabei sorgt die interne Beschaltung eines Vorschaltgerätes je nach Leuchtmittel dafür, dass die benötigte Spannung zur Verfügung gestellt oder der benötigte Zündimpuls erzeugt wird. Vorschaltgeräte werden in ein konventionelles Vorschaltgerät (KVG) und ein elektronisches Vorschaltgerät (EVG) unterschieden. Im Folgenden soll die Funktionalität der unterschiedlichen VGs kurz anhand von Beispielen erläutert werden.

KVGs wurden lange Zeit dafür verwendet, um Leuchtstofflampen oder auch Energiesparlampen zu betreiben. Heutzutage werden sie zunehmend durch EVGs ersetzt. Ein KVG besteht häufig im Wesentlichen aus einer Vorschaltrossel, die mit der Netzfrequenz arbeitet und die benötigte Zündspannung erzeugt und einem Starter. Durch Betätigen des Lichtschalters schließt sich der Stromkreis und der Starter sorgt mit seinen

internen Bauteilen für den Zusammenbruch des zuvor entstandenen Magnetfeldes an der Drossel. Durch den plötzlichen Zusammenbruch entsteht ein Hochspannungsimpuls der als Zündspannung für das Leuchtmittel dient und dieses zündet. Durch die im KVG verbauten Bauteile, wie unter anderem der ohmsche Widerstand der Wicklung kommt es zu Kupferverlusten, Wirbelströmen und Ummagnetisierungsverlusten. Es entstehen dabei etwa 22% Lampenleistungsverluste³.

EVGs betreiben das Leuchtmittel, anders als das KVG, mit wesentlich höheren Frequenzen von einigen kHz. Mithilfe von Gleichrichtern wandeln sie die Spannung zunächst zu einer Gleichspannung um und nutzen dann einen Wechselrichter um eine Wechselspannung mit höherer Frequenz zu erzeugen. EVGs sind im Vergleich zu KVGs verlustarmer, erreichen dabei einen Leistungsfaktor von $\lambda > 0,95$ ⁴ und weisen einen flackerfreien Betrieb auf. EVGs können gleichzeitig mehrere Leuchtmittel betreiben, sofern diese in Reihe geschaltet sind. Es gibt dabei für unterschiedliche Leuchtmittel auch unterschiedliche EVGs. Über die im nächsten Kapitel 2.5) erwähnten Schnittstellen lassen sich Leuchtmittel mit Hilfe von EVGs steuern und auch über verschiedene Dimmverfahren dimmen. Man unterscheidet hier zwischen analog dimmbaren EVGs, die mit einem Steuersignal wie z.B. 1-10V angesteuert werden und digital dimmbaren EVGs, die mit einem Bussignal wie z.B. DALI angesteuert werden.

Als LED-Treiber werden Vorschaltgeräte bezeichnet, die für LED-Technologie ausgelegt sind und diese mit Gleichspannung betreiben. Treiber arbeiten in der Regel als Konstantstrom- oder Konstantspannungsquelle. Durch den Anschluss von LED-Modulen an einem Treiber mit Konstantstromquelle fließt ein konstanter Strom durch die Leuchtdiode und betreibt diese. Dabei können, mit Hinblick auf die maximal erzeugbare Spannung des Treibers, mehrere Module in Reihe geschaltet werden. Treiber mit einer Konstantspannungsquelle benötigen für den Betrieb von LED-Modulen eine zusätzliche Stromregelung, damit der fließende Strom begrenzt wird. Ansonsten würde der Strom bis zur thermischen Zerstörung der LED ansteigen. Die Regelung könnte im einfachsten Fall mithilfe eines PWM-Signals umgesetzt werden.

³Quelle:http://ftp.stadtbetriebe.at/energieberatung/Downloads/Stromverbrauch/Beleuchtung_Betriebsgeraete.pdf

⁴Quelle:https://www.vossloh-schwabe.com/uploads/tx_espdfssearch/VS-Komponenten-Systeme_2014_DE_01/0244.pdf

2.5. Dimmen von Leuchtmitteln

2.5.1. Dimmschnittstellen

KNX ist ein in der Gebäudeautomation verwendeter digitaler Feldbus zur reinen Steuerung von unter anderem der Beleuchtung und wird hier nur der Vollständigkeit halber erwähnt. Der wesentliche Vorteil von KNX ist die mögliche Anbindung von jedem elektrischen Verbraucher. Durch Programmierung der Anschlüsse können diese dem Verbraucher angepasst werden. Zudem ist es möglich, Sensordaten über den Bus abzufragen, was zusätzliche Informationsverkabelung erspart. Nachteilig zur herkömmlichen Elektroinstallation sind hier sicherlich die zusätzlichen Anschaffungskosten, der erhöhte Platzbedarf und die Programmierarbeit zu nennen.

DALI ist wie KNX ein Bus der in der Gebäude-, aber auch Schiffsautomation eingesetzt wird. DALI selbst stellt ein genormtes Protokoll dar und ist in der Normenreihe IEC 62386⁵ erfasst. Lichttechnische Betriebsgeräte, die über eine digitale DALI-Schnittstelle verfügen, können, ähnlich wie bei KNX, über Kurzadressen einzeln angesteuert werden. Dabei findet ein bidirektionaler Datenaustausch statt. Die unterschiedlichen Zustände und Stati können dann über eine SPS abgefragt, visualisiert und geändert werden. Auch hier sind der zusätzliche Mehraufwand und die höheren Kosten als nachteilig anzusehen. Die Vorteile gleichen denen vom KNX-Bus.

Analoge **0-10V**-Schnittstellen finden seit je her ihre Verwendung in der Automation. Die passiven Netzteile erwarten dabei eine eingehende Steuerleistung. Durch die einfache Steuerung und das Verständnis dieser analogen Technik wird diese oft verwendet. Mit einem niedrigen Strom lässt sich die Steuerung über dünne Leitungen günstig integrieren und verbraucht damit kaum Platz. Nachteilig ist dabei die aufwändige Anbindung von Endgeräten in ein bestehendes System und die verteilte Datenerfassung, die nicht wie bei DALI auf einem Bus gesammelt werden kann.

Analoge **1-10V**-Schnittstellen arbeiten auch mit einer analogen Steuerung direkt am Endgerät. Dabei ist diese Schnittstelle aber aktiv und erzeugt die Steuerleistung, anders als die 0-10V-Vorschaltgeräte, selbst. Dazu muss das jeweilige Steuergerät in der Lage sein, den von den Netzteilen gelieferten Strom aufzunehmen und die Steuerspannung zu verringern. Die 1-10V-Schnittstelle hat dabei einen oftmals entscheidenden Vorteil gegenüber der 0-10V-Schnittstelle. Bei einem Spannungsausfall der Hauptstromversorgung z.B. eines Leuchtkreises würde ein 0-10V-Vorschaltgerät keinen Fehlerfall detektieren, da 0V in dessen Arbeitsbereich fallen würde. Anders ist dies bei einem 1-10V-Vorschaltgerät. Fällt dort die Spannungsversorgung der Steuerspannung aus, erzeugt das Vorschaltgerät mit der aktiven Schnittstelle 10V, sodass das Leuchtmittel zu 100% einge-

⁵Norm der International Electrotechnical Commission

schaltet wird. Liegt die Spannung bei 1V, so ist nur das Endgerät auf sein Minimum gesteuert. Um das Leuchtmittel komplett auszuschalten wird ein zusätzliches Relais oder ein zusätzlicher Schalter benötigt. Diese Schnittstelle wird deshalb oft in Beleuchtungskreisen verwendet.

2.5.2. Dimmverfahren

Puls-Weiten-Modulation

Die Puls-Weiten-Modulation wird dazu verwendet ein analoges Signal in ein digitales zu wandeln. Dazu erzeugt der Pulsweitenmodulator eine Sägezahn- bzw. Dreiecksspannung und ein zusätzlicher Regler eine Referenzspannung. Ein Komparator (siehe Kapitel 2.8.3 schaltet immer dann ein, solange die Referenzspannung größer als die Sägezahn- bzw. Dreiecksspannung ist. Abbildung 2.3 zeigt eine vereinfachte Puls-Weiten-Modulations (PWMs)-Schaltung.

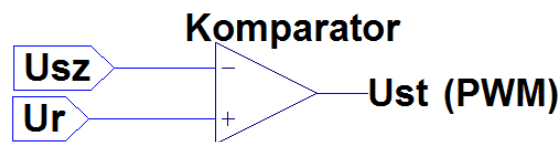


Abbildung 2.3.: Schaltung eines Pulsweitenmodulators

Wie in Abb. 2.4⁶ dargestellt vergleicht der Komparator die beiden Spannungen U_{SZ} und U_R und erzeugt das Ausgangssignal U_{St} mit dem Tastverhältnis

$$a = \frac{t_{ein}}{T} = \frac{U_R}{U_{SZ,max}} \quad (2.1)$$

wobei t_{ein} die Einschaltzeit und T die Periodendauer darstellen.

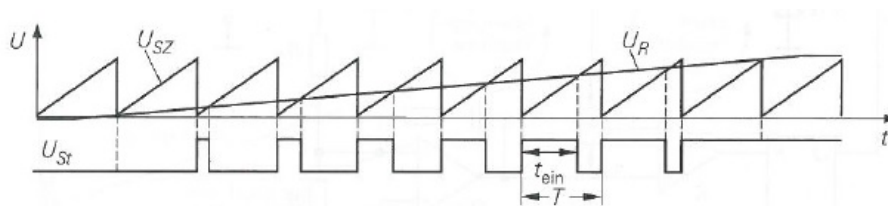


Abbildung 2.4.: Funktionsweise des PWMs

U_R ist dabei das analoge Eingangssignal und U_{St} das digitale Ausgangssignal, welches durch zwei Spannungsebenen (*LOW*- und *HIGH*-Pegel) gebildet wird.

⁶Quelle: [14], Seite 957, Abb. 16.48

In Bezug auf das Dimmverfahren anhand eines PWM-Signals lässt sich sagen, dass hierbei der Strom für das Leuchtmittel, vorzugsweise eine LED, periodisch ein- und ausgeschaltet wird. Ein flimmerfreier Verlauf erfolgt dabei ab einer Frequenz von 200Hz . Durch Einstellung der Einschalt- bzw. Ausschaltzeit während einer Periode wirken die hohen Frequenzen für das menschliche Auge so, als wäre das Licht gedimmt.

Phasen- / Phasenabschnittsteuerung

Als Phasensteuerung wird die Steuerung einer Wechselspannung bezeichnet. Nach jedem Nulldurchgang der Spannung wird diese bei einer bestimmten Phase an- oder abgeschnitten. Durch die Variation des Ein- bzw. Ausschaltwinkels lässt sich die Leistung an der Last verändern. Ein häufiger Anwendungsbereich ist die Helligkeitssteuerung von Leuchtmitteln.

Als wesentliches Bauelement wird in der Phasenabschnittsteuerung ein Triac und ein Diac verwendet.

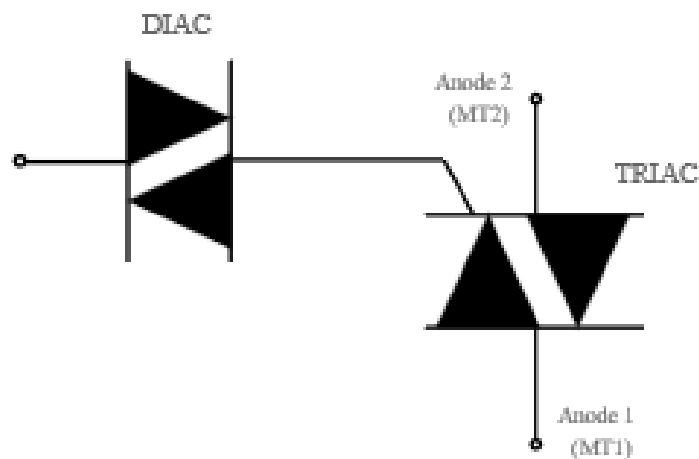


Abbildung 2.5.: Darstellung der Schaltsymbole eines Triacs (rechts) und eines Diacs (links)

Der Triac besteht aus einer Antiparallelschaltung zweier Thyristoren (siehe rechte Abb. 2.5⁷). Der Triac wird leitend, sobald das Gate mit einem positiven oder negativen Stromimpuls angesteuert wird. Hat ein Triac geschaltet, bleibt er bis zum nächsten Nulldurchgang des Stroms leitend. Andersherum bleibt er dann bis zum nächsten Gateimpuls sperrend. Ein Diac hingegen ist eine Zweirichtungs-Diode (vgl. linke Abbildung) und wird leitend, sobald dessen Durchbruchspannung überschritten wurde. Durch den bidirektionalen Aufbau können Diacs Wechselspannungen schalten.

⁷Quelle: https://www.electronics-notes.com/articles/electronic_components/scr/what-is-a-diac.php

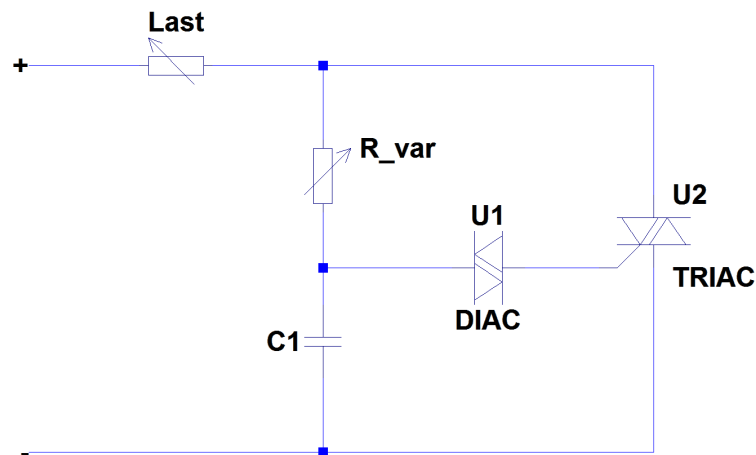


Abbildung 2.6.: Einfache Phasenanschnittsteuerung

Abbildung 2.6 zeigt eine einfache Dimmung über eine Phasenanschnittsteuerung. Dabei wird die Spannung an der Last im Grunde über das Potentiometer gesteuert. Durch einstellen des Widerstandes verschiebt sich die Phase. C_1 lädt sich abhängig vom eingestellten Widerstand auf und übertritt entsprechend der Zeitkonstanten RC die Schwellenspannung des Diacs, der nach seiner Zündung wiederum den Triac zünden lässt.

In Bezug auf das Dimmen über eine Phasen- / Phasenabschnittssteuerung lässt sich sagen, dass diese Technik bei der heutzutage weit verbreiteten LED-Technik weniger Anwendung findet. Sie wird oft in Vorschaltgeräten von Energiespar- und Leuchtstofflampen angewendet. Durch die Steuerung der Phasen im Wechselstromsystem müssen für die einphasige LED-Beleuchtung aufwändigere EVGs verwendet werden, die daraus eine Steuerung mit Gleichstrom erzeugen können.

In dieser Arbeit erfolgt demnach das Dimmverfahren nicht über eine Phasen- bzw. Phasenabschnittssteuerung.

2.6. Mikrocontroller

Mikrocontroller sind in der Industrie viel eingesetzte kleine elektronische Bauelemente und funktionieren mit ihrem On-Board-Speicher als ein Ein-Chip-Computersystem. Mikrocontroller gibt es in etlichen verschiedenen Bauformen und sind heutzutage in fast jeder elektronischen Steuerung und Regelung verbaut.

Ein Mikrocontroller besteht im Wesentlichen aus einem Prozessor (CPU⁸), einem Arbeitsspeicher (RAM⁹), einem Festspeicher (ROM¹⁰), einem Flashspeicher (EEPROM¹¹) und einer Peripherie¹².

Abbildung 2.7 zeigt beispielhaft einen Mikrocontroller der in der Industrie verwendet wird.



Abbildung 2.7.: Beispielhafter Mikrocontroller Atmel Mega328P

Der Atmel Mega328P ist ein Low-Power Mikrocontroller mit einem 8-Bit Prozessor, welcher mit 20MHz arbeitet und unter anderem 23 programmierbare I/Os, mehrere 10-Bit ADCs sowie eine I²C-, eine UART- und zwei SPI-Schnittstellen bietet.

Die modernen Mikrocontroller verfügen über interne Speicher wie z.B. den EEPROM auf welchen sie frei zugreifen können. Das Thema des internen Speichers wird an dieser Stelle nicht weiter vertieft. Mikrocontroller werden meist in den Programmiersprachen Assembler oder C programmiert, wobei Assembler bei Ressourcen armen Controllern verwendet wird. Die Hochsprache C wird immer dann verwendet, wenn der benötigte Speicher im Vergleich zu dem zur Verfügung stehenden Ressourcen gering ist. Mit

⁸CPU - Central Processing Unit

⁹RAM - Random Access Memory

¹⁰ROM - Read Only Memory

¹¹EEPROM - Electrically Erasable Programmable Read-Only Memory

¹²Peripherie - in der Informatik alle Geräte die an die Zentraleinheit (in diesem Fall dem Mikrocontroller) angeschlossen sind

C ist ein direkter Speicherzugriff und somit ein Hardware nahes Programmieren möglich.

Zusammenfassend ist es dem Mikrocontroller also möglich, eigene digitale und oder analoge Ein- und Ausgänge zu betreiben, eine Kommunikation mit anderen Geräten über eine der oben genannten Schnittstellen herzustellen und als Spannungsquelle zu dienen. Mit eigener Programmierung kann der Mikrocontroller dann auf die Anforderungen und benötigten Funktionen eingestellt werden.

2.7. Funknetze

Funknetze sind Netzwerke, in welchem Informationen in einem kabellosen Kommunikationssystem übertragen werden. Hierbei werden die in mehreren Normen unterteilten Funknetze wie z.B. Bluetooth und WLAN, klassifiziert.

Bei einer einzelnen Funkverbindung wird nicht mehr von Funknetzen gesprochen. Diese sind z.B. der Mobilfunk, Bündelfunk oder auch Rundfunk. Außerdem wird in der Kommunikation zwischen unidirektionalen, also in eine Richtung kommunizierenden und bidirektionalen, also in beide Richtungen kommunizierenden Verbindungen unterschieden. Ein Sender kann mit einem Unicast an einen Empfänger, mit einem Broadcast an alle möglichen Empfänger im Netz, oder mit einem Multicast an eine bestimmte Gruppe von Empfängern senden. Anwendungen finden Funknetze heutzutage in Sensornetzwerken, Heimnetzwerken (WLAN) oder auch Mobilfunknetzen.

Das MESH-Netzwerk ist eine besondere Art eines Netzwerks und erleichtert die redundante Auslegung, wodurch es an Kapazität und besonders an Zuverlässigkeit dazugewinnt. Das MESH-Netzwerk wird im Kapitel [2.7.4](#) noch einmal näher betrachtet.

Vor dem Hintergrund des IoT finden Funknetze immer mehr Anwendung. Durch sicherere und stabilere Funkverbindungen und kostengünstigere Technologien lassen sich kabelgebundene Informationskanäle durch kabellose Funkverbindungen ersetzen und erfüllen in der Regel die gleichen Anforderungen oder gewinnen zusätzliche Funktionen. Dabei wird der vorherige Hardwareaufwand vermehrt in die Software ausgelagert und die Informationsverarbeitung und -weiterleitung durch Programme übernommen.

2.7.1. Funkmodule

Unter einem Funkmodul wird in der Elektrotechnik ein Bauteil bezeichnet, welches als Funkempfänger und/oder -sender funktioniert. Es kann die zu sendenden Daten verschlüsseln und sichern und arbeitet dabei mit einem Funkprotokoll wie dem angesprochenen WLAN, ZigBee oder Bluetooth, indem es in dessen Frequenzband funkt. In Kombination mit z.B. einem Mikrocontroller können die Informationen der Datenkommuni-

kation verarbeitet werden. Dabei sind die Funkmodule herstellerunabhängig und arbeiten Funktechnologie-spezifisch.

2.7.2. Wireless Local Area Network (WLAN)

Das Wireless Local Area Network (WLAN) beschreibt ein kabelloses, lokales Netzwerk und ist eine Art eines Funknetzes. Das WLAN basiert dabei auf der IEEE-Norm 802.11 und arbeitet in der Schicht 1 und 2 des Open Systems Interconnection (OSI)-Schichtenmodells. Der aktuellste WLAN Standard IEEE 802.11ah wurde im Frühjahr 2016 eingeführt und arbeitet in einem Frequenzband von 750 bis 930MHz. Im Vergleich dazu arbeiteten seine Vorgänger noch im Gigahertz-Bereich. Dabei soll WLAN-ah durch Bündelung der Kanäle Datenraten bis zu $18 \frac{MBit}{s}$ erreichen¹³. Damit soll der WLAN Standard mit dem aktuellen Bluetooth LE, worauf hier nicht näher eingegangen wird und dem energiesparsamen ZigBee Standard konkurrieren.

2.7.3. ZigBee-Protokollstandard

In der Industrie wird der freie ZigBee-Protokollstandard für drahtlose Netzwerke beispielsweise in der Gebäude- und Hausautomation, Sensornetzwerken oder auch Beleuchtungssteuerung bereits verwendet. ZigBee selbst ist eine Spezifikation und beschreibt, welche funktionalen und nicht-funktionalen Anforderungen an das System erfüllt werden. Dabei baut ZigBee auf dem IEEE¹⁴ 802.15.4 auf und erweitert dessen Funktionalität.

In der Norm IEEE 802.15.4 wird ein Übertragungsprotokoll beschrieben. Dieses beinhaltet die untersten beiden Schichten des OSI-Modells, die Bitübertragungs- und die Sicherungsschicht. Dies entspricht dem Datenpaket-basierten Austausch zwischen zwei Geräten. Das ZigBee Protokoll setzt darauf auf und erweitert die Norm um die Schichten des Vermittler-, Sicherheits- und Anwendungslayer, wie in Abbildung 2.8 [13] gezeigt.

¹³Quelle: <https://www.elektronik-kompodium.de/sites/net/2101101.htm>; Online gesehen: 24.06.2018

¹⁴Institute of Electrical and Electronics Engineers

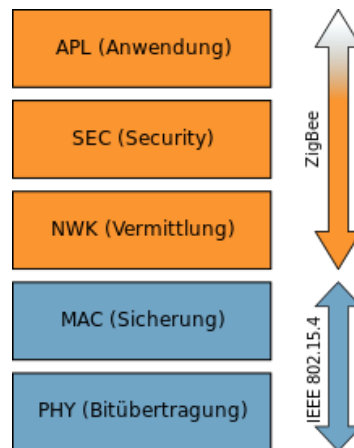


Abbildung 2.8.: ZigBee Protokollstack

Der Standard wird von der Organisation ZigBee-Alliance geführt und von ihr weiter entwickelt. Die ZigBee-Alliance [3] ist eine globale Mitgliederorganisation mit Sitz in den USA. Auf ihrer Homepage präsentiert sie aktuell die neueste Protokollversion ZigBee 3.0. Mit ZigBee 3.0 lassen sich laut technischer Spezifikation bis zu 65000 Geräte in einem Netzwerk aufnehmen. Das Protokoll, basierend auf der Norm IEEE 802.15.4-2011, sendet auf einer Frequenz von 2,4GHz und einer Datenrate von bis zu $250 \frac{kB}{s}$. Die Datenpakete werden mit einem AES-128 Blockverschlüsselungsalgorithmus verschlüsselt und gelten bis heute als kryptografisch gesichert [5].

2.7.4. MESH-Netzwerk

In einem allgemeinen vermaschten Netz ist jeder Netzwerkknoten mit einem oder mehreren anderen verbunden. Die Informationen werden dann von Knoten zu Knoten bis an die Zieladresse weitergeleitet. Fällt einer der Knoten aus, so wird die Information über einen anderen weitergeleitet und man spricht von einem selbst heilenden und redundanten Netzwerk. Ist jeder Knoten mit allen anderen verbunden spricht man hierbei von einem vollständig vermaschten Netz.

Im folgenden wird das MESH-Netzwerk anhand von Funkmodulen der Firma XBee 4.1.1 beschrieben und genauer erklärt.

Durch die Konfiguration mindestens eines Funkmoduls als Koordinator baut dieses unter Spannung ein ZigBee-MESH-Netzwerk mit der vorgegebene ID auf. Die anderen Funkmodule werden daraufhin eingefügt bzw. lassen sich von Koordinatoren oder Routern im selben Netzwerk einfügen (vgl. Kapitel 4.1.1). Die Besonderheit des MESH-Netzwerkes ist es, dass es selbst heilend ist. Das bedeutet, sollte einer der Knoten (Netzwerkteilnehmer) ausfallen, wird die Information über einen anderen Router bzw. Koor-

dinator an ihr Ziel weitergeleitet. Abbildung 2.9 zeigt ein beispielhaftes und vollständig vermaschtes Netzwerk, das über die Software XCTU visualisiert wurde.

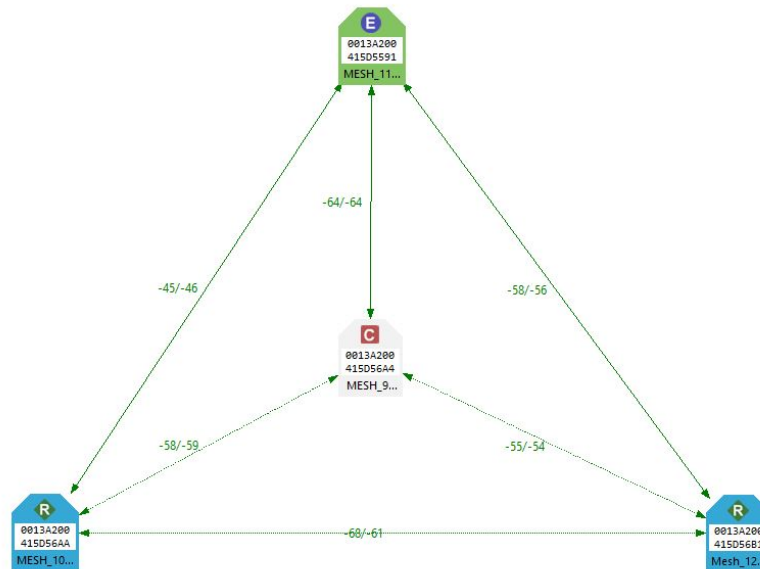


Abbildung 2.9.: Beispiel einer MESH-Map

Auf der Map ist zu sehen, dass dieses MESH-Netzwerk aus einem Koordinator, zwei Routern und einem End-Device besteht. Zwischen den Funkmodulen besteht jeweils eine Verbindung, die von der Software bewertet und angezeigt wird. Die Qualität der Verbindung entscheidet unter anderem darüber, wie schnell und fehlerfrei die Informationen weitergeleitet werden. Die Abbildung 2.10 zeigt die Referenztable, nach dem die Signalstärken der Verbindungen bewertet werden. Die Verbindungen werden in der Einheit dBm, Dezibel Milliwatt angegeben. Dezibel Milliwatt gibt das Verhältnis einer Leistung P im Vergleich zur Bezugsleistung von 1mW an und wird als Leistungspegel L_P bezeichnet. Der Leistungspegel wird dabei wie folgt berechnet:

$$L_P = 10 \cdot \log_{10}\left(\frac{P}{1\text{mW}}\right)$$

Für die Strecke des Koordinators zum End-Device besteht also eine Verbindung mit einem Leistungspegel von

$$\begin{aligned} P &= 10^{\frac{L_P}{10}} \cdot 1\text{mW} \\ &= 10^{\frac{-64\text{dBm}}{10}} \cdot 1\text{mW} = \underline{\underline{398,11\text{nW}}} \end{aligned}$$

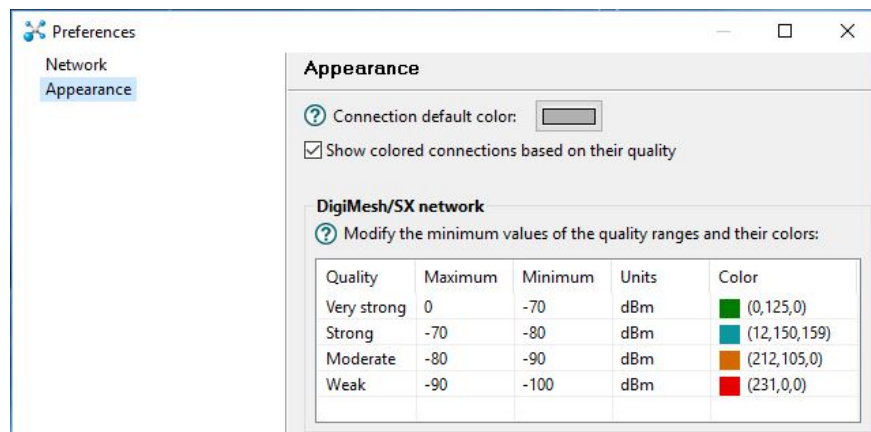


Abbildung 2.10.: Bewertungstabelle der Verbindungen der Software XCTU

Die Verbindungen können durch weitere Knotenpunkte verbessert werden, da dadurch die Wege zwischen ihnen verkürzt werden. Zwei Tests zur Überprüfung der Netzwerkqualität ergaben, dass zum einen die einfache Verbindung zwischen zwei Funkmodulen bis zu 20m Luftlinie (bei einem Pegel von rund -50dBm) in einem Bürogebäude ausreicht und zum anderen auch ungehindert an Bord von Schiffen und durch dessen Stahlwänden funktioniert. Letzteres zeigt die Abschlussarbeit von Herrn Marco Weirich (siehe Anhang A), welcher eine starke Verbindung von -74dBm (vgl. Abb. 2.10) zwischen zwei Modulen gemessen hat, die von einem Stahlschott getrennt waren. Arbeitet ein weiteres Netzwerk unmittelbar in der Nähe des Betrachteten, so hat dies keinerlei Auswirkungen auf die Kommunikation und die Qualität [10].

2.8. Theoretische Darstellung von grundlegenden Schaltungs- und Verfahrensvarianten

2.8.1. Super-Kondensatoren

Die allgemeine Beschreibung von Kondensatoren wird in dieser Arbeit als bekannt vorausgesetzt und nicht weiter erläutert. Anwendungsorientiert wird in diesem Fall detaillierter auf die Super-Kondensatoren eingegangen, die eine besondere Bauform des Kondensators darstellen.

Super-Kondensatoren werden je nach Ausführung ihrer Elektroden in drei unterschiedliche Familien unterteilt.

- *Pseudokondensatoren* besitzen Elektroden aus Metalloxiden und haben einen hohen Anteil von Pseudokapazität.

- *Doppelschichtkondensatoren* besitzen Elektroden mit einer sehr hohen statischen Doppelschichtkapazität. Der Anteil der Pseudokapazität ist nur gering.
- *Hybridkondensatoren* besitzen zwei unterschiedliche Elektroden. Eine mit einer hohen Doppelschicht-, die andere mit einer hohen Pseudokapazität.

Der Doppelschicht-Super-Cap besteht im Wesentlichen aus zwei Kollektoren, die jeweils eine polarisierte Elektrode darstellen. Im Zwischenraum befindet sich ein Elektrolyt, welches mit positiven und negativen Ionen gefüllt ist. Die Elektroden sind physisch von einem Separator getrennt. Abbildung 2.11¹⁵ zeigt eine schematische Darstellung des Lade- („charge“) und Entladevorgangs („discharge“).

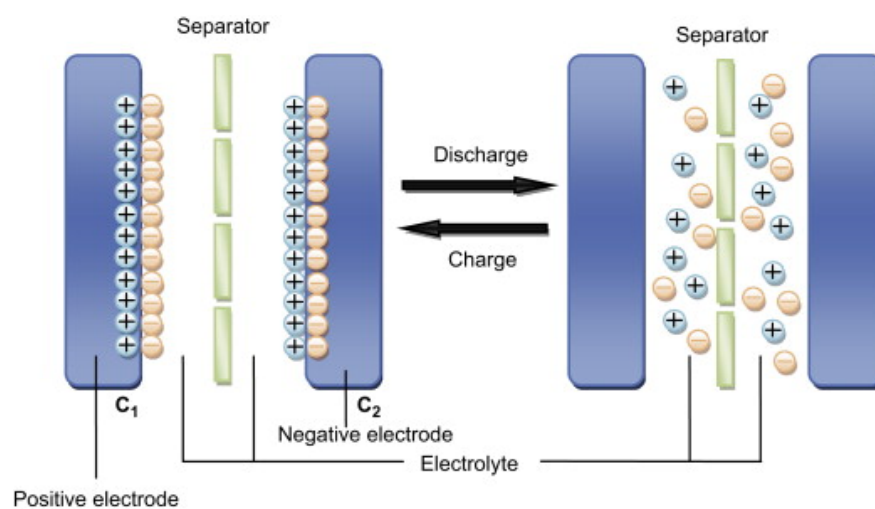


Abbildung 2.11.: Schematische Darstellung vom Laden und Entladen eines Doppelschicht-Super-Kondensators

Beim Anlegen einer Spannung an die Elektroden entsteht an dessen Rändern jeweils eine Helmholtz-Doppelschicht an der sich die positiven und negativen Ionen spiegelbildlich anordnen. Die Doppelschichten stellen im Ersatzschaltbild jeweils einen Kondensator C_1 und C_2 dar. Die Gesamtkapazität ergibt sich demnach aus deren Reihenschaltung.

Bei einem Pseudo-Super-Cap kommt es durch chemische reversible Redoxreaktionen an der Helmholtz-Doppelschicht dazu, dass Ionen aus dem Elektrolyt treten und in direkten Kontakt mit der Elektrode gelangen. Die dabei verfügbar werdenden Ladungsträger werden in einer so genannten Pseudokapazität gespeichert. Die Pseudokapazität ist ein Teil des elektrochemischen Kondensators und ergibt in Addition mit der Doppelschichtkapazität die Gesamtkapazität eines hybriden Super-Caps, welche in der Regel mehrere Farad betragen kann.

¹⁵Quelle: <https://energyeducation.ca/encyclopedia/Supercapacitor>

Super-Caps finden in der Industrie häufig dort eine Anwendung, wo große Energiemengen für kurze Zeit benötigt werden. Sie können dabei mehrere Hundert Ampere während kurzer Lastspitzen, Ströme im Milliampere-Bereich für einige Minuten oder μA über Monate und Jahre wie z.B. als Senderspeicher in Radios leisten.

2.8.2. Passive Filter

Passive Filter wurden allgemein dazu entwickelt, bestimmte Frequenzbereiche aus den unterschiedlichsten Signalen zu filtern. Man unterscheidet dabei in Tiefpassfilter, Hochpassfilter, Bandpass und Bandsperre. Passive Filter lassen sich, basierend auf Kombinationen von Widerständen R , Kondensatoren C und Spulen L realisieren.

Tiefpassfilter

In diesem Kapitel wird sich anwendungsorientiert ausführlicher mit dem RC-Tiefpass 1. Ordnung beschäftigt.

Der allgemeine Tiefpass lässt Frequenzen unter seiner Grenzfrequenz durch und filtert höhere. Abbildung 2.12 zeigt die Schaltung eines passiven Tiefpasses 1. Ordnung.

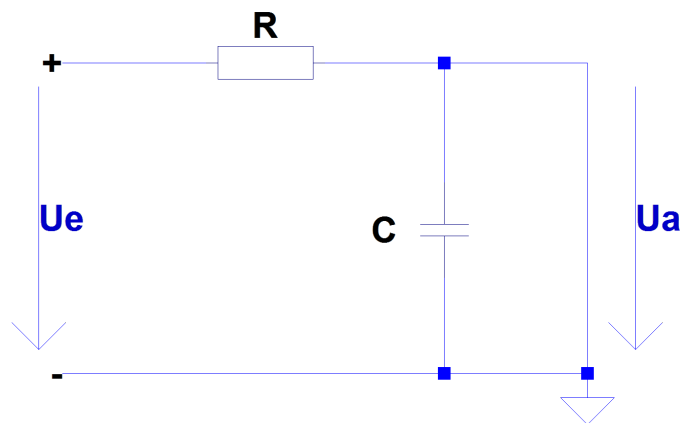


Abbildung 2.12.: RC-Tiefpass Schaltung

Die Übertragungsfunktion eines RC-Tiefpasses lässt sich mathematisch wie folgt beschreiben:

$$\underline{F}(j\omega) = \frac{\underline{U}_a(j\omega)}{\underline{U}_e(j\omega)} = \frac{1}{1 + j\omega RC} \quad (2.2)$$

Aus der Übertragungsfunktion 2.2 lässt sich nun die allgemein gültige Grenzfrequenz bestimmen:

$$\omega_g = 2\pi f_g = \frac{1}{RC} = \frac{1}{\tau} \quad (2.3)$$

$$\Rightarrow f_g = \frac{1}{2\pi RC} = \frac{1}{2\pi\tau} \quad (2.4)$$

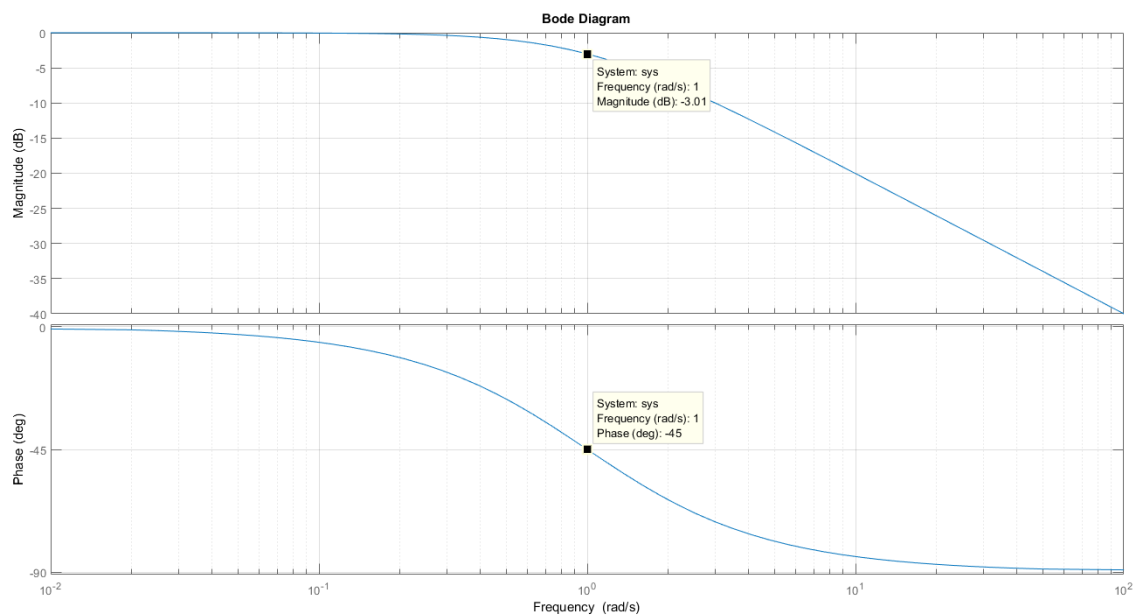


Abbildung 2.13.: Bodeplot eines RC-Tiefpass

Abbildung 2.13 zeigt den Bodeplot eines Tiefpasses 1. Ordnung und einer Grenzfrequenz von 1 Hz mit $R = 1000\Omega$ und $C = 1\text{ mF}$. Frequenzen die $< 1\text{ Hz}$ sind durchgelassen, $> 1\text{ Hz}$ zunehmend gefiltert. Durch Wahl der RC-Komponenten kann die Grenzfrequenz verschoben werden. Der Phasengang zeigt eine Phasendrehung von maximal $\varphi_{ges} = -90^\circ$ bei einer Phase von $\varphi = -45^\circ$ bei der Grenzfrequenz.

Hochpassfilter

Der Hochpassfilter lässt Frequenzen oberhalb seiner Grenzfrequenz annähernd unge- schwächt passieren und filtert tiefere. Der Hochpass 1. Ordnung setzt sich aus einer RC- Schaltung zusammen und kann mathematisch wie folgt beschrieben werden:

$$\underline{F}(j\omega) = \frac{\underline{U}_a(j\omega)}{\underline{U}_e(j\omega)} = \frac{j\omega RC}{1 + j\omega RC} \quad (2.5)$$

Im Bodediagramm steigt der Verlauf vom y-Achsenabschnitt mit $20 \frac{dB}{dec}$ und nähert sich nach Erreichen der Grenzfrequenz einem stationären Wert an. Die Grenzfrequenz lässt sich ebenfalls mit der Gleichung 2.4 berechnen.

Bandpassfilter

Der Bandpass ist ein Filter 2. Ordnung und besteht aus einer LCR-Schaltung. Sein Durch- lassbereich ist durch seine Bandbreite B um die Resonanzfrequenz f_0 definiert. Die Resonanz- oder auch Mittenfrequenz lässt sich wie folgt aus der oberen Grenzfrequenz f_H und aus der unteren Grenzfrequenz f_L berechnen:

$$f_0 = \sqrt{f_H \cdot f_L} \quad (2.6)$$

Die Bandbreite kann mittels Dämpfungsgrad D errechnet werden. Der Zusammenhang lautet:

$$B = 2 \cdot D \cdot \frac{\omega_0}{2\pi} = 2 \cdot D \cdot f_0 = 2 \cdot D \cdot \sqrt{f_H \cdot f_L} \quad (2.7)$$

Bandsperr

Die Bandsperr bildet den Gegensatz zum Bandpass und filtert alle Frequenzen die in- nerhalb seiner Bandbreite liegen. Frequenzen außerhalb dieser können passieren. Die Resonanzfrequenz und Bandbreite werden dabei wie bei dem Bandpassfilter mit den Gleichungen 2.6 und 2.7 berechnet.

2.8.3. Operationsverstärker

Es gibt unterschiedliche Arten von Operationsverstärkern, man unterscheidet sie in vier verschiedene Typen. Abbildung 2.14¹⁶ zeigt die vier Operationsverstärker die in einem Spannungs- bzw. Strom-Eingang und einem Spannungs- bzw. Strom-Ausgang unter- schieden werden. Die unterschiedlichen OPs werden dann wiederum in einen invertie- renden und einen nichtinvertierenden Verstärker unterteilt.

¹⁶Quelle: [14], Seite 511, Abb. 5.3

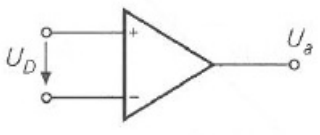
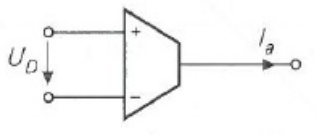
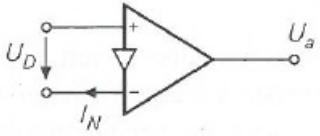
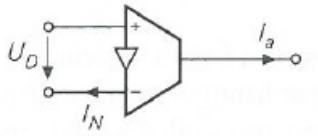
	Spannungs-Ausgang	Strom-Ausgang
Spannungs-Eingang	<p>Normaler OPV VV-OPV</p>  $U_a = A_D U_D$	<p>Transkonduktanz-Verstärker VC-OPV</p>  $I_a = S_D U_D$
Strom-Eingang	<p>Transimpedanz-Verstärker CV-OPV</p>  $U_a = I_N Z = A_D U_D$	<p>Strom-Verstärker CC-OPV</p>  $I_a = k_I I_N = S_D U_D$

Abbildung 2.14.: Schaltsymbole und Übertragungsgleichungen der vier Operationsverstärker

Normaler Operationsverstärker

Der „normale“ Operationsverstärker (OP) oder auch VV-OPV¹⁷ besitzt einen hochohmigen Eingang und ist der OP mit dem größten Marktanteil und wird in der Industrie am häufigsten verwendet.

Der Grundsatz einer OP-Schaltung besagt, „das sich die Ausgangsspannung eines Operationsverstärkers so einstellt, dass die Eingangsspannungsdifferenz Null wird¹⁸“.

Abbildung 2.15 zeigt beispielhaft die Schaltung eines nicht invertierenden Operationsverstärkers.

¹⁷V = Voltage; Spannungssteuerung am Eingang (V) - Spannungsquelle am Ausgang (V) - Operationsverstärker (OPV)

¹⁸sinngemäß [14], Seite 515

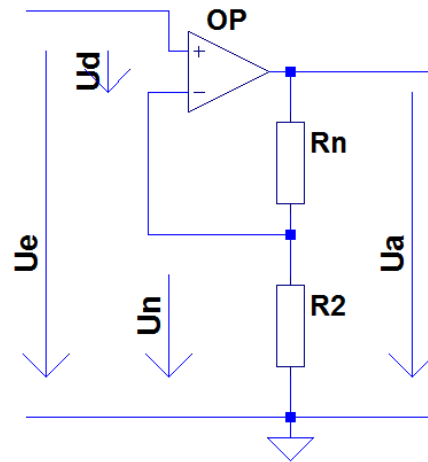


Abbildung 2.15.: Schaltung eines nicht invertierenden Operationsverstärkers

Daraus lässt sich die Ausgangsspannung U_a wie folgt berechnen:

$$U_a = A_D \cdot U_D \quad (2.8)$$

Für den nicht invertierenden Verstärker gilt zudem:

$$V_U = A = 1 + \frac{R_N}{R_1} \quad (2.9)$$

Transkonduktanz-Verstärker

Der Transkonduktanz-Verstärker oder auch VC-OPV¹⁹ besitzt hochohmige Ein- sowie Ausgänge. Der Ausgang verhält sich dabei wie eine Stromquelle, die durch die Eingangsspannungsdifferenz U_D gesteuert wird. Der Ausgangsstrom errechnet sich dabei wie folgt:

$$I_a = S_D \cdot U_D$$

mit der Differenzsteilheit als Ableitung von I_a nach U_D im Arbeitspunkt

$$S_D = \left. \frac{dI_a}{dU_D} \right|_{AP}$$

Der Begriff der Transkonduktanz²⁰ kommt dadurch, weil das Verhalten dieses Verstärkers durch die Steilheit S_D bestimmt wird.

¹⁹V = Voltage; C = Current; Spannungssteuerung am Eingang (V) - Stromquelle am Ausgang (C) - Operationsverstärker (OPV)

²⁰Transkonduktanz = Übertragungsteilheit

Der Transkonduktanz-Verstärker wird häufig bei der Umsetzung von spannungsgesteuerten Lautstärkereglern oder auch Modulatoren verwendet.

Transimpedanz-Verstärker

Der Transimpedanz-Verstärker oder auch CV-OPV²¹ besitzt einen stromgesteuerten invertierenden Eingang und eine Spannungsquelle am Ausgang. Die Bezeichnung erhält der Verstärker aufgrund seiner internen charakteristischen Impedanz Z , die im Megaohm-Bereich liegt.

Abbildung 2.16 zeigt die beispielhafte Schaltung eines Transimpedanz-Verstärkers mit dem Rückkopplungswiderstand R . U_a kann je nach Vorzeichen des Stromes negativ oder positiv sein und lässt sich wie folgt mathematisch beschreiben:

$$U_a = A_D \cdot U_D = -I \cdot R \quad (2.10)$$

Bei positiven Strom auf den invertierenden Eingang des OPs muss dieser in der Lage sein eine negative Ausgangsspannung erzeugen zu können, ansonsten kann die OP-Schaltung nicht in diesem Spannungsbereich arbeiten.

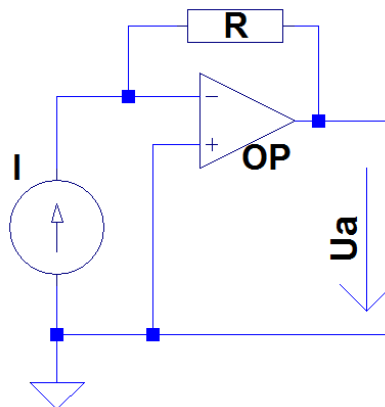


Abbildung 2.16.: Schaltung eines Transimpedanz-Verstärkers

Der Transimpedanz-Verstärker wird unter anderem in der Messtechnik eingesetzt, um präzise Messungen kleiner Ströme durch die Verstärkung zu ermöglichen.

²¹C = Current; V = Voltage; Stromsteuerung am Eingang (C) - Spannungsquelle am Ausgang (V) - Operationsverstärker (OPV)

Strom-Verstärker

Der Strom-Verstärker oder auch CC-OPV²² besitzt einen stromgesteuerten Eingang und einen stromgesteuerten Ausgang. Der Ausgangsstrom I_a stellt sich wie folgt ein

$$I_a = S_D \cdot U_D = k_I \cdot I_N$$

und wird durch die Steilheit S_D gesteuert. Der Ausgangsstrom lässt sich zudem auch über den Stromübertragungsfaktor

$$k_I = \left. \frac{dI_a}{dI_N} \right|_{AP}$$

als Ableitung des Ausgangsstroms I_a nach dem Eingangstrom I_n im Arbeitspunkt ermitteln.

Der Strom-Verstärker wird oft dazu verwendet bekannte Grundschaltungen wie Subtrahierer oder Integrierer zu bilden.

Komparator

Der Komparator ist eine besondere Anwendung eines OPs, die zwei Eingangsspannungen miteinander vergleicht. Sobald die Spannung am nicht-invertierenden Eingang höher ist als die Spannung am invertierenden Eingang nähert sich die Ausgangsspannung der positiven Versorgungsspannung. Umgekehrt geht die Ausgangsspannung gegen die negative Versorgungsspannung.

Der Komparator lässt sich aus jedem beliebigen OP realisieren. Das besondere Merkmal ist dabei die fehlende Rückkopplung. Die Abbildung 2.17²³ zeigt das Schaltsymbol eines Komparators.

²²C = Current; Stromsteuerung am Eingang (C) - Stromquelle am Ausgang (C) - Operationsverstärker (OPV)

²³Quelle: <https://www.efxkits.us/lm324-comparator-ic-working-applications/>

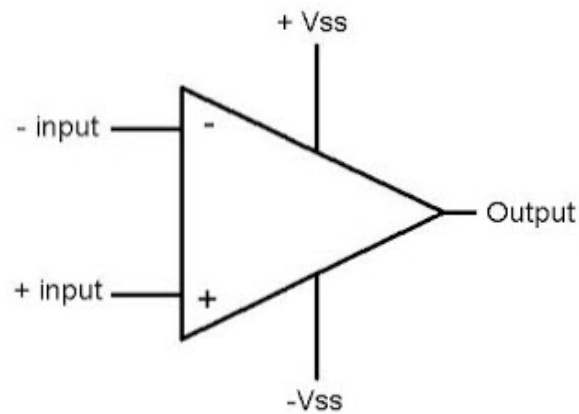


Abbildung 2.17.: Schaltsymbol eines Komparators

Häufig wird der Komparator dazu verwendet ein analoges Signal zu digitalisieren und wird z.B. für die Spannungsüberwachung anhand eines Schwellenwertes genutzt. Außerdem werden Komparatoren dazu verwendet PWM-Signale zu erzeugen (siehe Kapitel [2.5.2](#)). Auch in einem Analog-Digital-Umsetzer werden unter anderem mehrere Komparatoren verwendet, um ein analoges Signal mit einer bestimmten Auflösung zu digitalisieren. Dabei ist die Höhe der Auflösung abhängig von der Anzahl der Komparatoren.

3. Anforderungen und Konzepte

3.1. Anforderungen an das System

In dieser Arbeit soll eine kabellos-kommunizierende Beleuchtungssteuerung entwickelt und getestet werden. Die Beleuchtungssteuerung soll auf bestehenden Technologien aufsetzen und bekannte Dimmschnittstellen und Vorschaltgeräte verwenden.

Dabei sollen die Kabelwege von den jeweilige Leuchten zu den Tastern eingespart und diese mit Batterien betrieben werden.

Die Batteriekapazitäten sollen aufgrund von Wartungszwecken bei Betätigung des Tasters überwacht werden.

Die Leuchten sollen über Versorgungskabel vom 230V AC/50Hz -Bordnetz gespeist werden.

Das verwendete Leuchtmittel soll, wenn möglich, aus LED-Technik bestehen und mit einer Dimmschnittstelle aus dem Kapitel 2.5 ansteuerbar sein.

Es soll eine Anbindung von Sensoren an die Leuchtenmodule möglich sein. Zudem sollen die Messwerte durch das eigene System oder Drittsysteme genutzt werden können. Zum Zweck der Darstellung der Nutzbarkeit werden die Messwerte auf einer externen SPS der FA Bachmann angezeigt.

Es soll ein Notbetrieb realisiert werden in den das System bei einem detektierten Spannungsausfall einer Versorgung übergeht. Dieser soll den Vorschriften der Klassifikationsgesellschaften entsprechen.

Für die Anbindung einer Schiffsautomation sollen die Zustände der Leuchten und die Messwerte der Sensoren über eine Modbus-RTU-Schnittstelle abrufbar sein. Zusätzlich soll ein benutzerrechteabhängiger Zugriff auf die Beleuchtungssteuerung über die Schnittstelle umgesetzt werden. Die Funktionalität soll anhand einer grundlegenden Visualisierung nachgewiesen werden.

Für die technische Umsetzung dieser Arbeit sollen prototypische Schaltungen entworfen

werden, die die zuvor genannten Funktionen ermöglichen und die notwendigen Schnittstellen anbieten.

3.2. Konzept für die Schalt- und Versorgungsplatine der Leuchtenmodule

3.2.1. Konzept für die Spannungsversorgung der Platine

Für die in dieser Arbeit entwickelte Platine der Leuchtenmodule wird ein 230V AC Eingang benötigt. Die Spannung muss im Weiteren auf eine Gleichspannung transformiert werden, um die verwendeten Bauteile zu versorgen. Am Ausgang werden die 230V AC wieder ausgegeben und an das EVG der Leuchte angeschlossen. Zur Sicherheit sollen zwei Massepotentiale für die Gleich- und Wechselspannung definiert, um die unterschiedlichen Spannungsebenen zu trennen.

3.2.2. Konzept für das Schalten der Leuchten

Wird der Befehl zum An- bzw. Ausschalten durch kurzes Betätigen des Lichtschalters an das Leuchtenmodul gesendet, sollen die 230V AC am Ausgang der Platine ausgegeben werden, sodass die Leuchte einschaltet.

Die Leuchtenmodule sollen spezifischen Tastermodulen zugewiesen, so dass Lichtszenarien realisiert werden können.

3.2.3. Konzept für das Dimmen der Leuchten

Für das Dimmen der Leuchten sind die im Kapitel 2.5 aufgeführten Dimmschnittstellen und Verfahren etabliert. Es soll ein flimmerfreier und kontinuierlicher Dimmvorgang realisiert werden.

Wird der Befehl zum hoch oder runter dimmen durch langes Halten des Tasters an das Leuchtenmodul gesendet, wird die verwendete Schnittstelle über einen Mikrocontroller derart angesteuert, dass sich das Leuchtmittel kontinuierlich zwischen seinem minimalen Wert und 100% einstellt.

3.2.4. Konzept für den Anschluss von Sensoren an der Platine

Für den Anschluss der Sensoren wird ein Mikrocontroller verwendet. Hierzu sollen exemplarisch die Verwendung der analogen und digitalen Schnittstellen gezeigt werden. Wenn möglich werden die Sensoren mit der gleichen Versorgungsspannung arbeiten können, die der Mikrocontroller benötigt, um weitere Spannungstransformation zu vermeiden.

Die Auswahl der Sensorik soll anwendungsorientiert sein und einen Mehrwert im Praxiseinsatz bieten. Für die Überwachung der Alterung des Leuchtmittels oder der Raumbeleuchtungsstärke empfiehlt sich ein Lichtstärkesensor sowie ein Temperatursensor für die Temperaturüberwachung im Leuchtenkörper oder des Raumes (siehe Kapitel 7).

3.3. Konzept für die Schalt- und Versorgungsplatine der Tastermodule

3.3.1. Konzept für die Spannungsversorgung der Platine

Das Tastermodul soll unabhängig vom Bordnetz sein und kabellos versorgt werden. Dazu soll eine kapazitive Energiequelle verwendet werden. Die Energiequelle soll möglichst platzsparend und leicht zu warten sein.

Dafür bietet sich für den Prototypen ein Batteriepack an, welches zusätzlich mit einer Spannungsüberwachung überwacht werden kann.

3.3.2. Konzept für die Steuerungsbefehle an die Leuchtenmodule

Durch kurzes oder langes Betätigen des Tasters soll das Tastermodul einen Befehl generieren, welcher die Informationen zum Schalten bzw. Dimmen beinhaltet. Dieses soll dann an alle Leuchtenmodule versendet werden, die zum jeweiligen Tastermodul gehören.

3.4. Konzept für die Web-Visualisierung

Für die Web-Visualisierung soll eine Schnittstelle zwischen der seriellen UART Kommunikation und dem Modbus-RTU hergestellt werden, damit die Informationen über Schalterzustände der Leuchten, Sensordaten, usw. der Schiffsautomation zur Verfügung gestellt werden können. Zusätzlich soll ein benutzerrechteabhängiger Zugriff auf die Beleuchtungssteuerung umgesetzt werden. Die Schnittstelle soll außerdem dazu verwendet werden eine eigene Visualisierung auf einer SPS der FA Bachmann zu erstellen. Hierfür wird eine Hardware-Schnittstelle und eine Softwareumsetzung sowie eine lizenzierte SPS benötigt.

4. Realisierung der Konzepte

Alle Datenblätter der in dieser Arbeit verbauten und benannten Bauteile sind in den Anhängen hinterlegt.

4.1. Definition der Datenübertragung und Datenverarbeitung

4.1.1. Beschreibung vom XBee S2C Funkmodul

Die *XBee S2C*-Funkmodule der Firma DIGI¹ (siehe Abb. 4.1²) arbeiten mit dem beschriebenen ZigBee-Protokoll aus Kapitel 2.7.3 und bietet damit eine Hardware-technische Lösung zur Realisierung von MESH-Funknetzen, die mit dem standardisierten Protokoll arbeiten.



Abbildung 4.1.: XBee SC2 Funkmodul der Firma DIGI

Die Module sind über die hauseigene Software XCTU konfigurierbar oder lassen sich über eine beliebige Terminal Software per AT-Befehl³ konfigurieren. Darüber lassen sich die Funkmodule so programmieren bzw. parametrieren, dass diese MESH-Netzwerke aufbauen und in ihnen Daten versenden bzw. empfangen. Dabei kann jedes Funkmodul als eine der folgenden drei Gerätearten betrieben werden:

¹<https://www.digi.com/>

²Quelle: <https://www.digi.com/products/models/xb24cdmpit-001>

³AT - Attention

Router:

Router können sich über Anmeldung bei einem Router, welcher sich in dem gewünschten Netzwerk befindet, einfügen. Nach dem Beitritt erhält das Gerät eine 16-Bit-Kurzadresse vom hinzufügenden Router. In einem MESH-Netzwerk, also einem Netzwerk in welchem jeder Netzwerkknoten mit einem oder mehreren anderen verbunden ist, erfolgt die Adressvergabe zufällig. Ein Router ist befugt Pakete durch das Netzwerk zu verteilen und zu versenden.

Koordinator (Coordinator):

Durch Versorgung eines Koordinators öffnet dieser ein Netzwerk mit der ihm vorgegebenen ID. Andere Geräte können sich nun in das Netzwerk einwählen. Koordinatoren haben in bestehenden Netzwerken die gleichen Funktionen wie ein Router.

Endgerät (End Device):

End Devices stellen Geräte wie z.B. Sensormodule dar. Sie können ZigBee-Netzwerken hinzugefügt werden, haben aber nicht alle Funktionen zur Verfügung. Sie können z.B. erhaltene Datenpakete nicht weiter im Netzwerk verteilen. End Devices haben die Möglichkeit in einen Schlafmodus zu gehen, was einen Batteriebetrieb dieser Geräte ermöglicht.

Anbinden lassen sich die Funkmodule über deren digitale und analoge Schnittstellen. Abbildung 4.2 zeigt die Pinbelegung des verwendeten Funkmoduls.

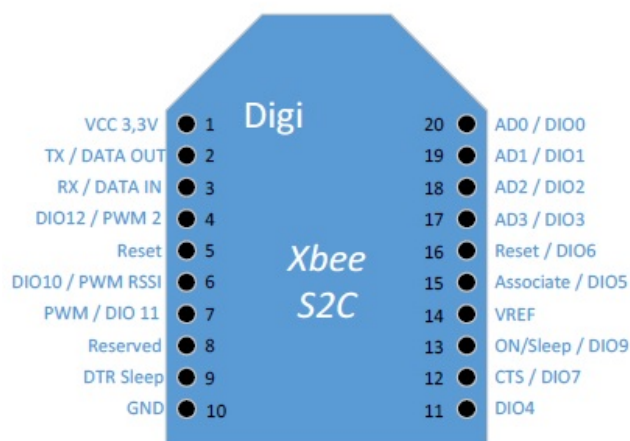


Abbildung 4.2.: Pinbelegung des XBee SC2 Funkmoduls

Das Modul wird über den V_{cc} -Pin mit 3,3VDC versorgt. Über die Pins Tx^4 und Rx^5 lässt sich eine serielle Universal Asynchronus Receiver Transmitter (UART) Kommunikation mit dem Modul aufbauen, worüber dieses unter anderem parametrisiert wird. Zudem lässt sich mit dem Modul eine Kommunikation über eine Serial Peripheral Interface (SPI)-Schnittstelle herstellen. Neben weiteren digitalen und analogen I/O-Pins besitzt der XBee S2C einen Pin für den Schlafmodus. Dieser Pin überwacht die anliegende Spannung auf Pegeländerungen, solange sich ein End Device im Schlaf- bzw. Energiesparmodus befindet. Zusätzlich kann das Modul z.B. über Pin 4 oder 6 ein PWM Signal erzeugen. Weiter verwendete Pinfunktionen des Funkmoduls werden im Laufe dieser Arbeit genauer beschrieben.

Aufgrund seiner kompakten und kleinen Bauweise, der geringe Leistungsaufnahme und den passenden Betriebsarten sowie Funktionen, wird dieses Produkt mit ZigBee als Funkstandard für die technische Umsetzung dieser Arbeit ausgewählt. Ausschlaggebend war die Möglichkeit mit den Modulen ein MESH-Netzwerk aufbauen zu können. Zudem ist es mit einem Entwicklungsboard der gleichen Firma erweiterbar und unter anderem mit mehreren Mikroprozessoren programmier- und bedienbar. Zusätzlich ist das Funkmodul und der Aufbau eines MESH-Netzwerkes bei WSAM bereits verwendet und getestet worden [15].

Alternativ wäre es möglich gewesen das Funknetz über WLAN mit entsprechenden Funkmodulen zu realisieren. Aufgrund des wesentlich höheren Energieverbrauchs ist eine batteriebetriebene Lösung aber nicht realisierbar. Eine alternative über Bluetooth ist ebenfalls nicht umsetzbar, da mit Bluetooth kein MESH-Netzwerk umgesetzt werden kann. Ebenso die Near Field Communication (NFC), die zu geringe Reichweiten aufweist. Lediglich das Kommunikationsprotokoll Z-Wave würde einen Marktstandard aus der Hausautomation bieten, der eine technische Umsetzung ermöglichen könnte. Aufgrund der zuvor genannten Aspekte werden diese Alternativen jedoch verworfen.

4.1.2. Beschreibung vom Atmel Mega328P

Der ATmega328P ist ein verbrauchsarmer Mikrocontroller (vgl. Kapitel 2.6) der Firma Atmel und wird unter anderem auf den Boards der Firma Arduino verbaut.

Das Mikrocontrollerboard Arduino Pro Mini wird mit der Arduino eigenen Software „Arduino IDE“ in einer C-derivaten Sprache programmiert. Der Vorteil liegt dabei in der Nutzung der C-Standardbibliotheken und zusätzlicher Möglichkeiten objektorientiert zu programmieren. Ein wesentlicher Nachteil der Entwicklungsumgebung ist, dass das Board über die IDE nicht zu debuggen ist, was vor allem die Fehlersuche erschwert. Für Anwendungsfälle mit hohem Programmieraufwand beinhaltet das einen hohen

⁴transmit - senden

⁵receive - empfangen

Aufwand bei der Fehlersuche.

Über die Pins *Tx* und *Rx* lässt sich der Mikrocontroller, genau wie das Funkmodul, beschreiben und programmieren. Neben der Spannungsversorgung über die Pins *VCC* und *GND* hat der Wattuino Pro Mini mehrere digitale und analoge I/O-Pins, wovon wiederum mehrere Pins ein PWM-Signal liefern können. Am Ausgang liefert er 3,3V mit maximal 150mA Ausgangsstrom in Summe (vgl. Datenblatt im Anhang A).

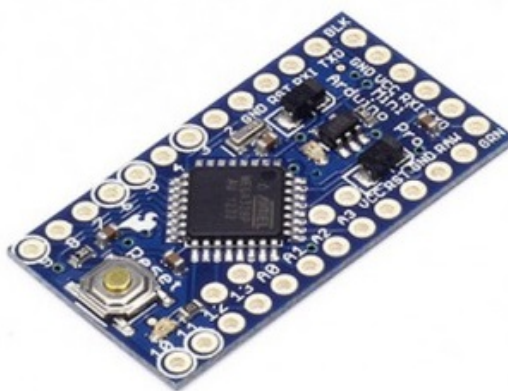


Abbildung 4.3.: Arduino Pro Mini 3,3V/8MHz

Das in Abb. 4.3⁶ gezeigte, günstige und auf dem Markt weit verbreitete Mikrocontroller-Board mit dem Atmel Mega328P bietet alle benötigten digitalen und analogen Ein- und Ausgänge und verfügt über die nötigen Kommunikationsschnittstellen wie USB und UART. Nach der Installation eines Bootloaders ist das Board kompatibel mit den Arduino-Bibliotheken. Ein wesentlicher Aspekt ist, dass das kompakte und verbrauchsarme Board in der oben genannten Ausführung genau wie das XBee-Modul mit 3,3V DC versorgt wird.

Alternativ wäre eine Verwendung des Low Power Mikrocontrollers Atmel Tiny85 möglich, welcher aber nicht die gleichen Kommunikationsschnittstellen wie der Atmel Mega328P zur Verfügung stellt und sich somit nicht für die Umsetzung dieser Arbeit eignet. Auch der MSP430FR2311 der Firma Texas Instruments stellt eine Alternative dar. Der Mikrocontroller würde die gleichen Anforderungen, wie der Atmel Mega328P erfüllen. Durch die wesentlich höheren Anschaffungskosten wird aber auch diese Alternative verworfen.

⁶Quelle: <https://store.arduino.cc/usa/arduino-pro-mini>

4.2. Entwicklung von Printed Circuit Boards

Anforderung dieser Arbeit war es Prototypen auf Steckboard-Ebene zu entwickeln. Um eine platzsparende und technisch übersichtliche Lösung umzusetzen wurden jedoch Printed Circuit Boardys (PCBs) mithilfe der Software *KiCad* entworfen. Die industriellen Platinen beinhalten die Schaltungen zu den einzelnen Funktionen wie Abb. 4.4 und Abb. 4.13 zeigen. Die AC und DC Spannungsebenen wurden durch unterschiedliche Massen getrennt. Die Leiterbahnen wurden dabei auf beide Seiten verteilt. Die einzelnen Bauteile wurden daraufhin auf die Platinen gelötet. Die kompletten Schaltpläne sind dem Anhang A zu entnehmen.

4.3. Schalt- und Versorgungsplatine der Leuchten

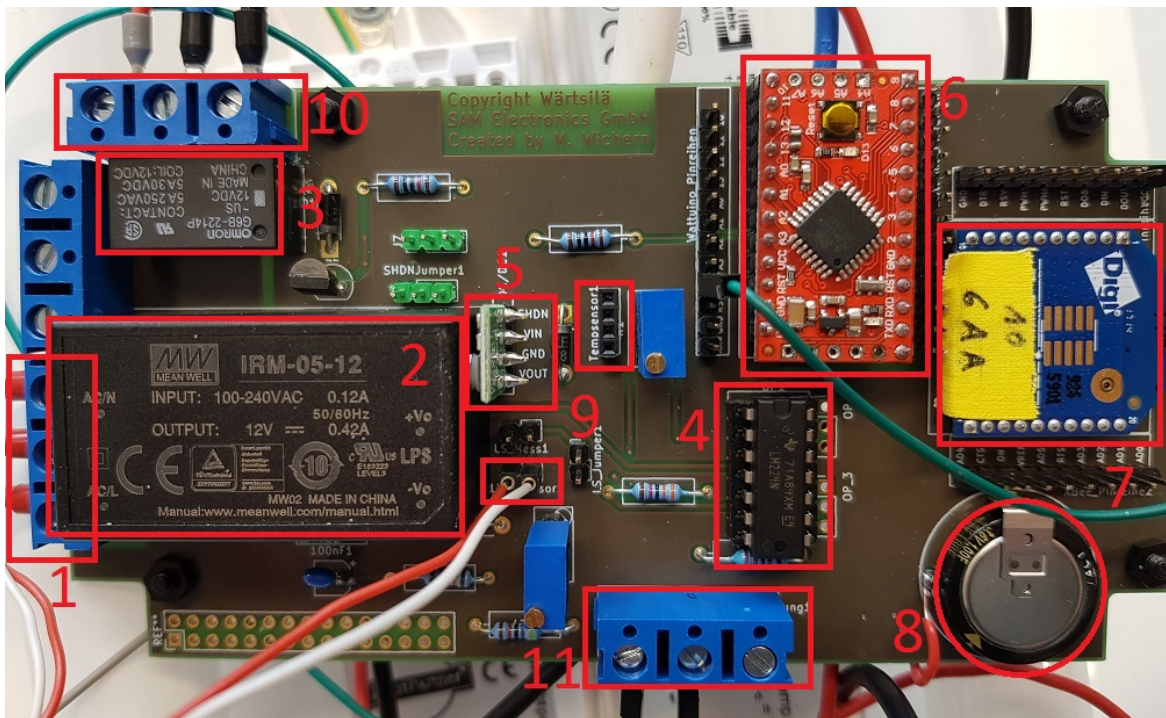


Abbildung 4.4.: Gesamte Platine der Leuchtenmodule

Die Platine wird über ein Versorgungskabel mit 230V AC versorgt. Die Adern werden auf den 3-poligen Klemmblock in Abb. 4.4 (1) auf der linken Seite gelegt. Die Spannung wird dann durch das Netzteil der Firma Meanwell IRM-05-12 (2) auf 12V DC und einem Ausgangsstrom von 0,42A transformiert. Die Spannung wird benötigt, um das Omron G6B-2214P Relais (3) und den Operationsverstärker LM224N (4) zu speisen. Durch den

Pololu Step-Down Wandler (5) wird die Spannung von 12V DC auf 3,3V DC transformiert, um den Mikrocontroller Atmel Mega328P (6), das XBee-Funkmodul (7), die Super Kondensatoren (8) und die Sensoren (9) zu versorgen.

Die 230V AC werden über den 3-poligen Klemmblock in der oberen linken Ecke der Platine in Abb. 4.4 (10) an das EVG der Leuchte ausgegeben. Zusätzlich wird die 1-10V Dimmschnittstelle aus dem unteren Klemmblock (11) über eine zweiadrige Steuerleitung an das EVG gelegt.

4.3.1. Beschreibung der verwendeten Schaltungen

Schaltung zum Schalten der Leuchtmittel

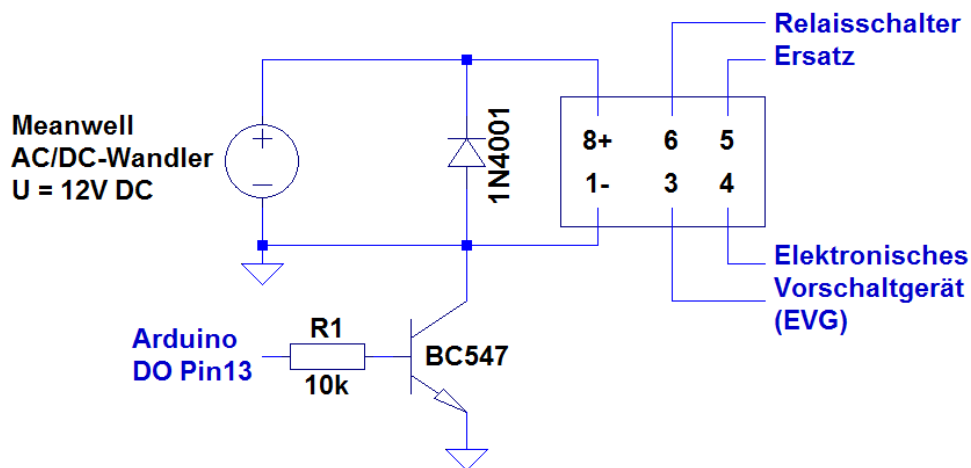


Abbildung 4.5.: Schaltplan zum Schalten der Leuchtmittel

Wird einem erhaltenen ZigBee-Protokoll entnommen, dass der Befehl zum Schalten des Leuchtmittels erteilt wurde, geht der digitale Output-Pin 13 des Atmel Mega328P auf *HIGH*, also 3,3V und arbeitet als Quelle für den Transistor *BC547*. Für einen Kollektorstrom von 16,7mA, den das Relais benötigt um sauber zu schalten, braucht man bei einem Verstärkungsfaktor $B = 200$ (siehe Datenblatt im Anhang A) einen Basisstrom von min. $83,5\mu\text{A}$ um den Transistor sicher durchzusteuern.

Bei der Wahl eines $10\text{k}\Omega$ Widerstandes fließt gemessen ein Basisstrom von $218\mu\text{A}$, was ausreicht, um das Relais zu schalten. Der Transistor wirkt nun wie ein geschlossener Schalter, so dass am Relais 12V DC anliegen. Daraufhin schalten die beiden Kontakte 3 und 4, sodass am Ausgangsklemmblock 230V AC anliegen und an das EVG geschaltet werden. Gleichzeitig schalten die Kontakte 5 und 6, welche als Ersatz auf einen 3-poligen

Klemmblock gelegt werden. Um den Transistor zu schützen wird eine 1N4001 Freilaufdiode in Sperrrichtung parallel zum Relais angeschlossen.

Schaltung zum Dimmen der Leuchtmittel

Für das Dimmen im Rahmen dieser Arbeit wird eine 1-10V Schnittstelle verwendet, die in diesem Fall mit einer OP-Schaltung umgesetzt wird. Der verwendete OP ist einer von vier OPs des LM224N. Die Schaltung ist EVG unabhängig und nutzt das PWM-Signal des analogen Output-Pins 3 vom Atmel Mega328P, um die Schnittstelle zu steuern. Das PWM-Signal des Output-Pins stellt in Reihe mit dem $R_1 C_1$ -Tiefpass eine analoge Spannungsquelle (3,3V) mit einem gemessenen Spannungspegel von 0V bis 2,96V dar.

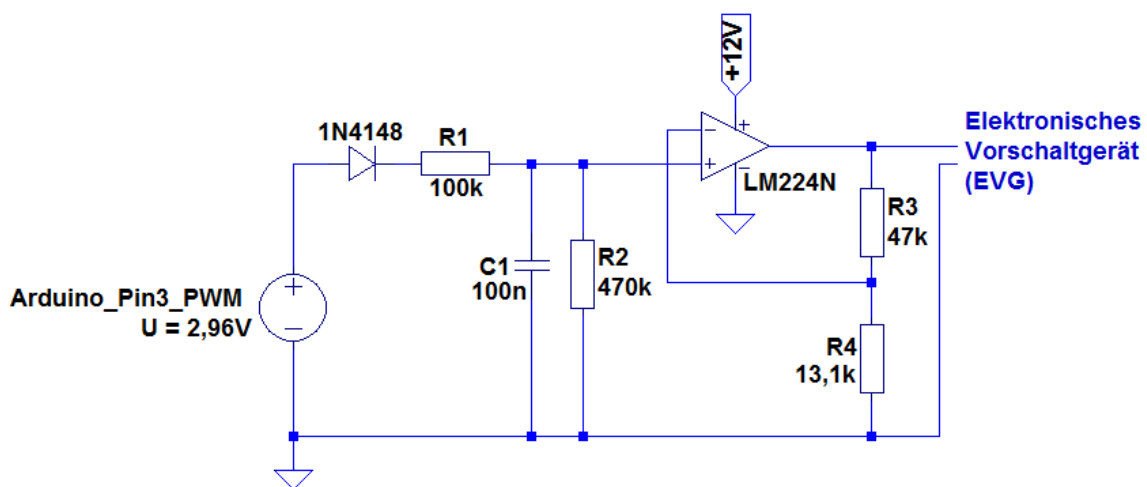


Abbildung 4.6.: Operationsverstärkerschaltung zur Dimmung eines beliebigen 1-10V EVGs über ein PWM-Signal

Aus der Gleichung 2.4 erhält man für diese Schaltung eine Grenzfrequenz von

$$f_c \approx \frac{1}{2\pi \cdot R_1 \cdot C_1} = \frac{1}{2\pi \cdot 100k\Omega \cdot 100nF} = \underline{\underline{15,92Hz}}$$

was für $R_2 \gg R_1$ gilt. Die Bauteile sind dabei so gewählt, dass die Dämpfung die Frequenzen $> f_g$ filtert. Um den Kondensator des Tiefpasses sicher entladen zu können, sobald die Spannung vom Arduino 0V beträgt, wird ein zusätzlicher Widerstand R_2 parallel zum Tiefpass geschaltet. Dieser wird benötigt, da am hochohmigen Eingang des OPs nur sehr wenig Strom fließt. Damit die Grenzfrequenz nicht erhöht wird, wird R_2 hochohmig gewählt.

Aufgrund der geringen PWM-Frequenz von gemessen 250Hz filtert der Tiefpass nicht das gesamte Rauschen auf dem Signal. Um dem entgegenzuwirken wird zusätzlich eine

1N4148-Diode in Sperrrichtung zwischen dem Arduino und dem Tiefpass geschaltet (vgl. Abb. 4.6). Das hat außerdem zur Folge, dass sich der Strom über den Widerstand R_2 entlädt und nicht zurück in den Arduino fließt.

Dabei fallen an der Diode, dem $R_1 C_1$ -Tiefpass und dem Widerstand R_2 gemessen 0,8V Spannung ab, sodass am Eingang des OPs 2,16V anliegen. Zum Steuern der Schnittstelle muss der OP nun die Spannung auf 10V am Ausgang verstärken. Dabei fallen im Bauteil gemessen 1,4V ab, sodass die erforderliche Versorgungsspannung gerundet 12V beträgt. Berücksichtigt man die inneren Bauteilverluste nicht, würde der OP seine Ausgangsaussteuerbarkeitsgrenze überschreiten (vgl. Anhang A) und am Ausgang würden keine 10V mehr anliegen. Dafür wird der Spannungsteiler und die Verstärkung V_U aus Gleichung 2.9 mit einer Eingangsspannung $U_e = 2,16V$ wie folgt bestimmt.

$$\begin{aligned}U_e \cdot V_U &= U_a = 10V \\V_U &= \frac{10V}{2,16V} \\1 + \frac{R_3}{R_4} &= \frac{10V}{2,16V}\end{aligned}$$

Unter der Annahme $R_3 = 47k\Omega$ erhält man

$$R_4 = \frac{R_3}{\frac{10V}{2,16V} - 1} \approx \underline{\underline{13,1k\Omega}}$$

Mit der nun entwickelten Schaltung lässt sich ein beliebiges EVG mit einer 1-10V Schnittstelle dimmen.

Schaltung zum Anbinden der Sensoren

Wie im Abschnitt 3.2.4 beschrieben wird im Rahmen dieser Arbeit ein analoger Lichtstärkesensor (Eigenbau aus einer BPW 34 Photodiode) und der digitale Temperatur- und Luftfeuchtigkeitssensor DHT11 / MAKEVMA311 an die Platine angebunden. Abbildung 4.7 zeigt den Schaltplan für die Sensorik.

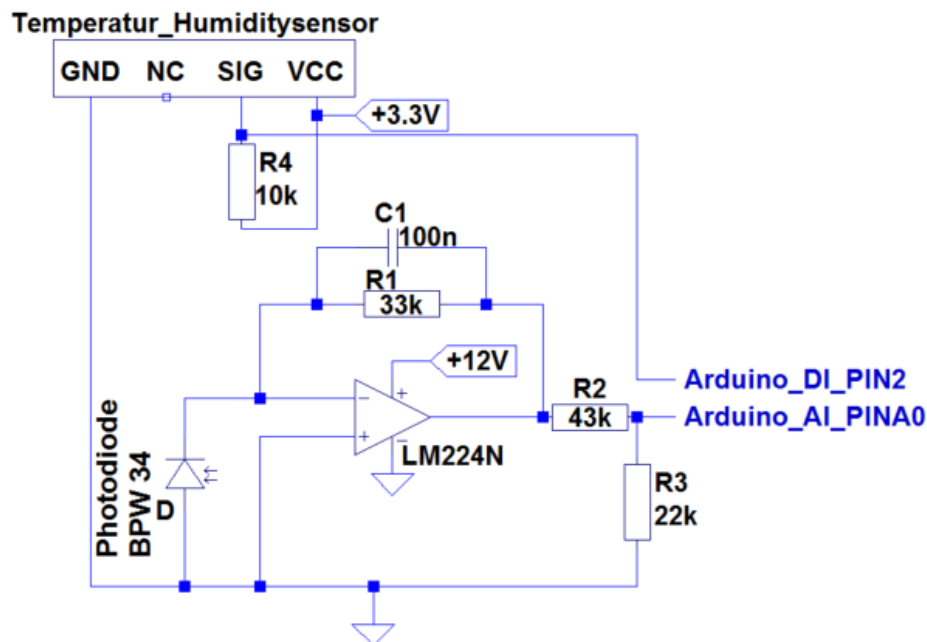


Abbildung 4.7.: Schaltplan der angebundenen Sensorik

Der digitale Sensor wird mit der gleichen Spannung wie der Mikrocontroller und XBee mit $VCC = 3,3V$ und $GND = 0V$ versorgt. Der Sensor misst die Werte analog und konvertiert diese mit einem Sensorboard internen Integrated Circuit (IC) in ein digitales Protokoll (siehe Abb. 4.8). Die Datenleitung *SIG* wird auf den digitalen Input-Pin 2 vom Arduino gelegt und darüber ausgelesen. Der zusätzliche Widerstand R_4 dient als Pull-Up Widerstand zu VCC . Durch Einbinden der Bibliothek *dht.h* lassen sich die Temperatur und die Luftfeuchtigkeit über Abfragebefehle vom Mikrocontroller auslesen. Der Pin *NC* steht für „Not Connected“.

40-Bit Packet

Integral Humidity Read	Decimal Humidity Read	Integral Temperatur Read	Decimal Temperatur Read	Checksum
8-Bit	8-Bit	8-Bit	8-Bit	8-Bit

Abbildung 4.8.: Schematische Darstellung des Datenprotokolls des digitalen Sensors

Als analoger Sensor wird ein Lichtstärkesensor verwendet der aus einer BPW 34 Photodiode (siehe Datenblatt im Anhang A) umgesetzt wird. Im Leerlauf hat die Photodiode laut Datenblatt eine Leerlaufspannung von 350mV. Zudem weist sie einen linearen Fotostrom von $80\mu A$ bei einer Lichtstärke von $E_v = 1000\text{Lux}$ auf. Die Diode wird in

diesem Fall mithilfe eines Transimpedanz-Verstärkers (siehe Kapitel 2.8.3) mit einem OP vom LM224N in einen quasi Kurzschluss gebracht. Um den negativen Strom der Diode auszugleichen und möglichst $U_d = 0V$ zu erreichen, fließt ein positiver Strom durch die Rückkopplung und erzeugt so eine Spannung an R_1 . Um Schwingungen auf dem rückgekoppelten Signal zu filtern wurde experimentell ein 100nF Kondensator ermittelt der parallel zum Rückkopplungswiderstand vorgesehen wird (vgl. Abb. 4.7). Der hier verwendete LM224N hat den Vorteil, das er mit einer Single Supply versorgt wird. Dies vereinfacht die Spannungsversorgung erheblich. Er kann dadurch zwar keine negativen Spannungen erzeugen, was in der vorliegenden Schaltung jedoch auch nicht benötigt wird. Nach internen Bauteilverlusten liegen nun maximal 10V an R_1 an. Durch den zusätzlichen Spannungsteiler R_3 und R_4 stellt sich am analogen Input-Pin A0 eine Spannung von 0V bis 3,3V ein, die in einen Analogwert von $\frac{1023}{3,3V} = 310 \frac{1}{V}$ gewandelt wird und interpretiert werden muss.

Dafür wurde eine Messreihe des Sensors aufgestellt und unter selbst geschaffenen Laborbedingungen (siehe Abb. 4.9) mit dem Luxmeter *testo 540* kalibriert.

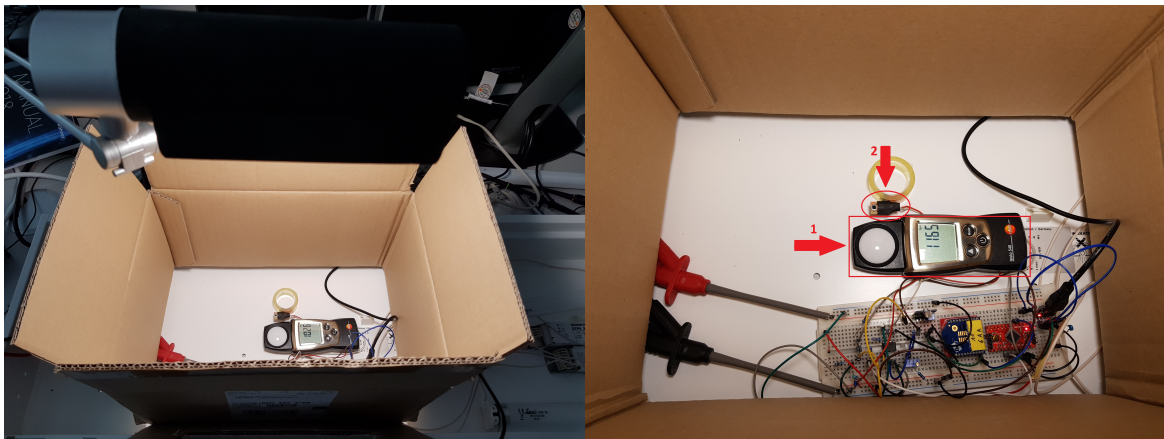


Abbildung 4.9.: Kalibrierung des analogen Lichtstärkesensors (Markierung 2, rechtes Bild) mit dem Luxmeter *testo 540* (Markierung 1, rechtes Bild)

Tabelle 4.1.: Messreihe zur Kalibrierung des Lichtstärkesensors mit dem Luxmeter *testo 540*

Lfd. Nr.	Lampenhöhe [cm]	Messwert Luxmeter [Lux]	analoger Messwert Photodiode $\frac{1\text{Lux}}{\text{V}}$	Faktor $\frac{\text{Luxmeter}}{\text{Photodiode}}$ [1]
1	-	0	0	0
2	62	560	8,6	65,116
3	49	1040	17,2	60,465
4	41,5	1665	31,6	52,69
5	32	2055	39	52,692
6	30	2280	41,4	55,073
7	34	2620	51,5	50,874
8	26	3000	59,7	50,251
9	24	4155	84,5	49,172
10	20	4970	96,6	51,449
11	17	6080	112,1	54,237
12	14	7075	133,2	53,116

¹ Mittelwert über 10 Messwerte

In der Tabelle 4.1 wurden die Ergebnisse der Messreihe zur Kalibrierung festgehalten. Für den durchschnittlichen Faktor f erhält man nach den Messergebnissen aus der Tabelle

$$\begin{aligned}
 f &= \frac{\sum_{n=2}^{11} \text{Faktor}}{\text{Anzahl der Messwerte}} \\
 &= \frac{595,135}{11} = \underline{\underline{54,103}}
 \end{aligned}$$

Der Faktor $f = 54,103$ wird nun im Programm des Atmel Mega328P nach dem Auslesen der Sensormesswerte verrechnet, um einen aussagekräftigen Messwert in der Einheit *Lux* zu erhalten.

Es kann, ohne Beachtung der Exaktheit dieses analogen Sensors, gezeigt werden, dass die Anbindung und Auswertung eigener analoger Sensorik möglich ist. In Bezug auf die Exaktheit wurde bei Messungen mit anderen Leuchtmitteln unterschiedliche Faktoren festgestellt. Wahrscheinliche Ursachen dafür sind die unterschiedlichen Lichtspektren der Leuchtmittel (siehe Kapitel 2.1). Das hat zur Folge, dass die mittlere Beleuchtungsstärke die von der Fotodiode gemessen wird von der spektralen Empfindlichkeit der Diode abhängt (vgl. Abb. 4.10). Die Ausarbeitung eines universell einsetzbaren Lichtsensors ist nicht Anforderung dieser Arbeit gewesen. Für exakte Messwerte müssen der Messung

die spektrale Empfindlichkeit des Sensors und das jeweilige Spektrum des zu messenden Leuchtmittels hinterlegt sein.

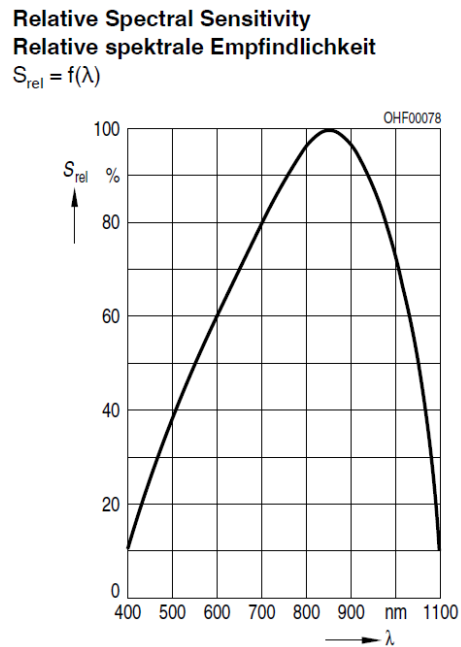


Abbildung 4.10.: spektrale Empfindlichkeit der BPW 34 Fotodiode

Schaltung für den Notbetrieb

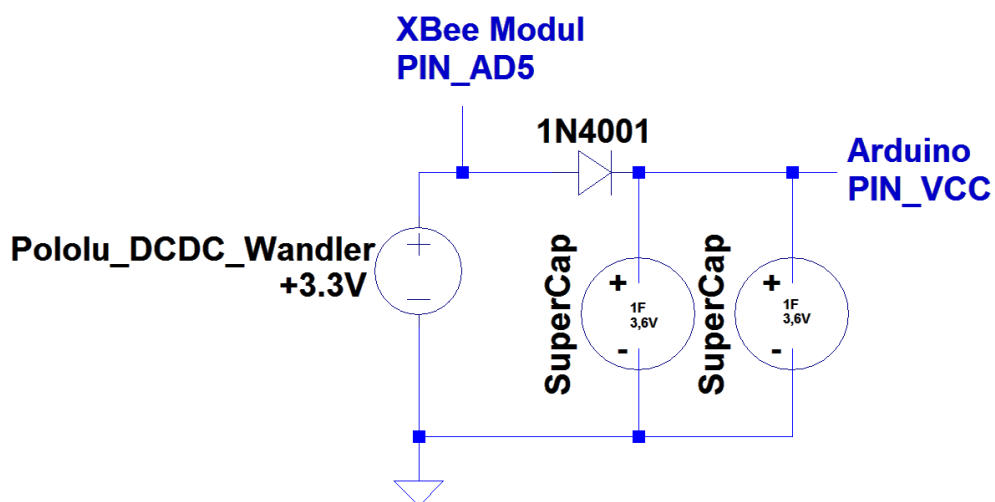


Abbildung 4.11.: Schaltplan für den Notbetrieb

Abbildung 4.11 zeigt den Schaltplan für den Notbetrieb. Bei Detektion eines Spannungsausfalls vom XBee Pin AD5 vor der 1N4001 Diode sendet dieser den Befehl an alle Leuchten in seinem zugewiesenen Netzwerk auf 100% Leuchtkraft einzuschalten. Die beiden parallel zur Spannungsversorgung geschalteten Super-Kondensatoren (vgl. Kapitel 2.8.1) versorgen den Atmel Mega328P und das XBee Modul bis zu 9s, damit diese einen Notbetrieb-Befehl versenden und ihren aktuellen Schaltzustand und Dimmwert im Arduino-internen EEPROM-Speicherbaustein speichern können.

Abbildung 4.12 zeigt die Zeitmessung der Versorgung durch die Kondensatoren. Zunächst werden diese ab dem Zeitpunkt (1) durch Zuschalten des Netzes geladen. Zum Zeitpunkt (2) wird die Versorgung durch das Netz getrennt, sodass sich die Kondensatoren entladen und die Schaltung, insbesondere den Mikrocontroller und das XBee-Modul versorgen. Zum Zeitpunkt (3) bei 1,9V liegt die minimale Versorgungsspannung der Bauteile (siehe Datenblätter im Anhang A), deshalb wurde die Zeitspanne bis dorthin gemessen und beträgt ungefähr 9s. Zum Zeitpunkt (4) sind die Kondensatoren vollständig entladen.

EDU-X 1002A, CN57350547: Mon Jul 30 21:52:40 2018

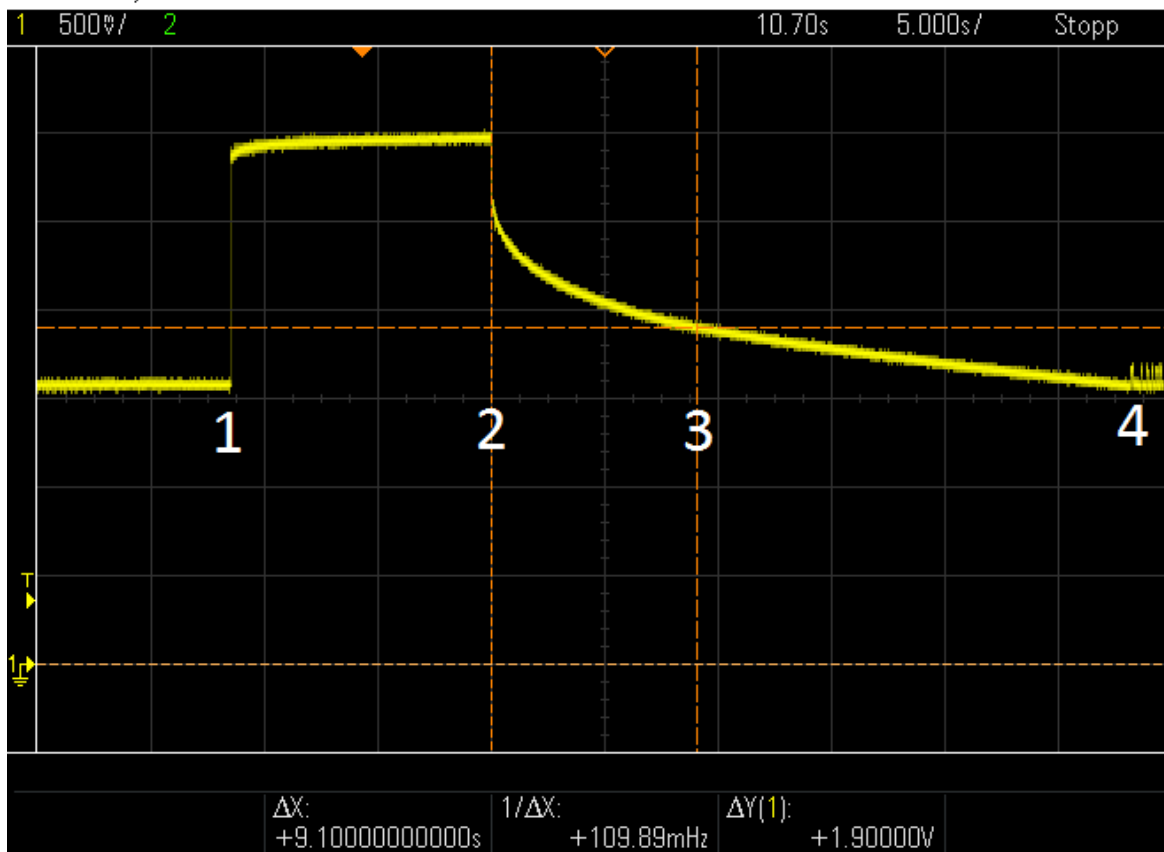


Abbildung 4.12.: Zeitmessung der Versorgung durch die Kondensatoren

4.4. Schalt- und Versorgungsplatine der Taster

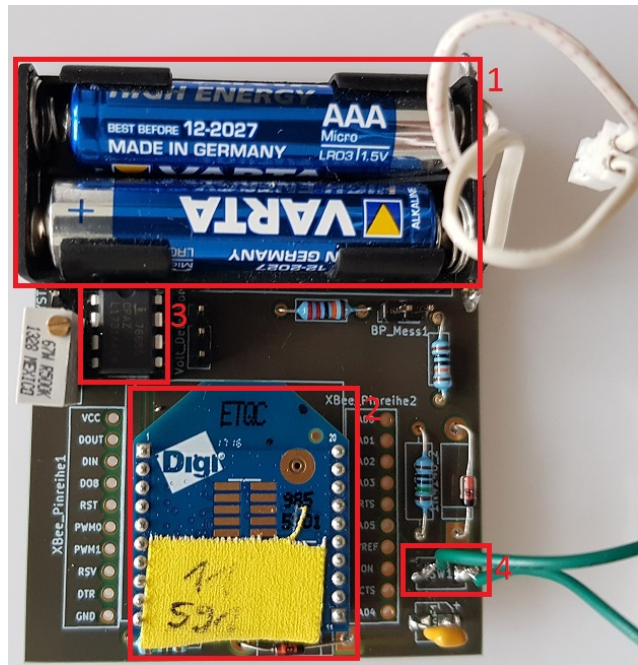


Abbildung 4.13.: Gesamte Platine der Tastermodule

Die Platine wird über das Batteriepack (1) durch zwei AAA Micro Batterien mit insgesamt 3V versorgt. Das Batteriepack versorgt damit das XBee Modul (2) und den Spannungsüberwachungschip ICL7665 (3). Der Taster zum Schalten der Leuchtenmodule ist über zwei Adern auf der Platine (4) befestigt.

4.4.1. Schaltung zum Senden des Schaltbefehls

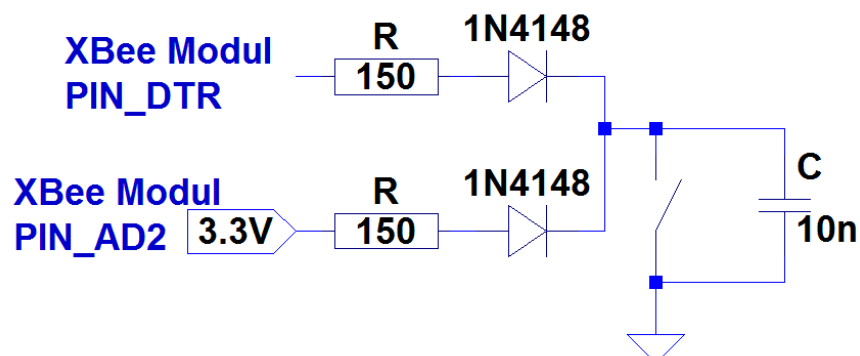


Abbildung 4.14.: Schaltplan zum Senden eines Schalt- oder Dimmbefehls

Für das Senden eines Schaltbefehls an die Leuchtenmodule wird der Taster *Switch* im Schaltplan (Abb. 4.14) betätigt. Dadurch wacht das XBee-Modul über den Pin DTR aus seinem energiesparenden Sleep-Modus auf und sendet ein Protokoll. In diesem Protokoll ist der Zustand des digitalen Ausgangs-Pins AD2 vom XBee-Modul entscheidend. Wird der Taster gedrückt liegen die 3V Versorgungsspannung an ihm und der Pin AD2 geht auf *HIGH*. Wird der Taster losgelassen oder gar nicht betätigt ist der Pin AD2 *LOW*. Dabei sendet das XBee-Modul jedes mal ein Protokoll, wenn sich der Zustand vom Pin AD2 ändert. Die Zustände von AD2 und die Zeit zwischen dem Erhalten der Protokolle werden dann vom Mikrocontroller in den Leuchtenmodulen interpretiert und ausgewertet.

4.4.2. Schaltung für die Batterieüberwachung

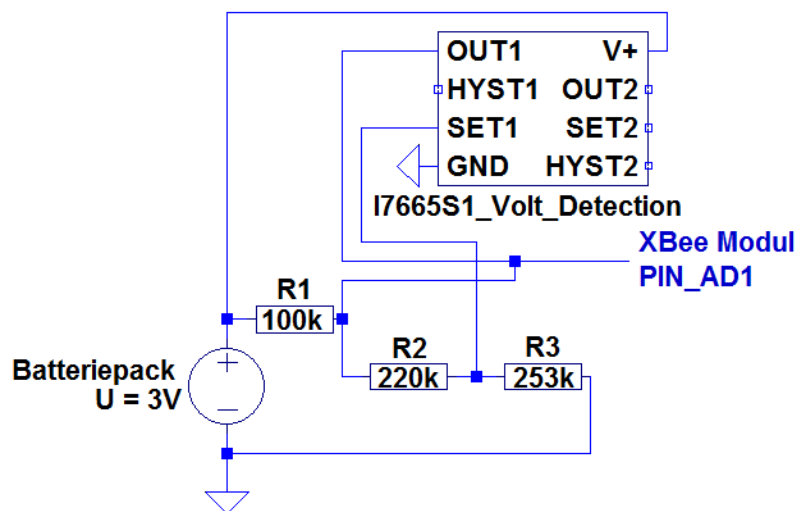


Abbildung 4.15.: Schaltplan der Batterieüberwachung

Abbildung 4.15 zeigt die Schaltung für die Batterieüberwachung der Tastermodule. Dafür wird der Spannungsüberwachungschip ICL7665 verwendet, welcher zwei interne Komparatoren und eine Referenzspannung von 1,3V besitzt. Mit einem davon vergleicht der IC seine interne Referenzspannung mit der Spannung *SET1*. Durch Auslegung der Widerstände $R_2 = 220k\Omega$ und $R_3 = 253k\Omega$ liegen bei einer Eingangsspannung von 3V, 1,6V an *SET1*. Mit sinkender Eingangsspannung sinkt auch die Spannung an *SET1* proportional. Das Resultat des Komparators wird über *OUT1* an den digitalen Eingangspin AD1 vom XBee gelegt.

Liegt die Spannung des Batteriepacks über 2,4V und somit $V_{SET1} > 1,3V$, ist der Pin AD1 = *LOW*. Die Batterieladung wird als ausreichend angezeigt.

Liegt die Spannung des Batteriepacks unter 2,4V und somit $V_{SET1} < 1,3V$, ist der Pin

$AD1 = HIGH$. Die Batterieladung wird als nicht mehr ausreichend angezeigt.

Dabei arbeitet das XBee-Modul laut Datenblatt bis 2,1V und Laborversuchen bis 1,9V Versorgungsspannung. Um einen Ausfall des Moduls zu verhindern, wird dem Betreiber so rechtzeitig signalisiert, dass die Batterien getauscht werden müssen.

EDU-X 1002A, CN57350547: Mon Jul 30 21:40:35 2018

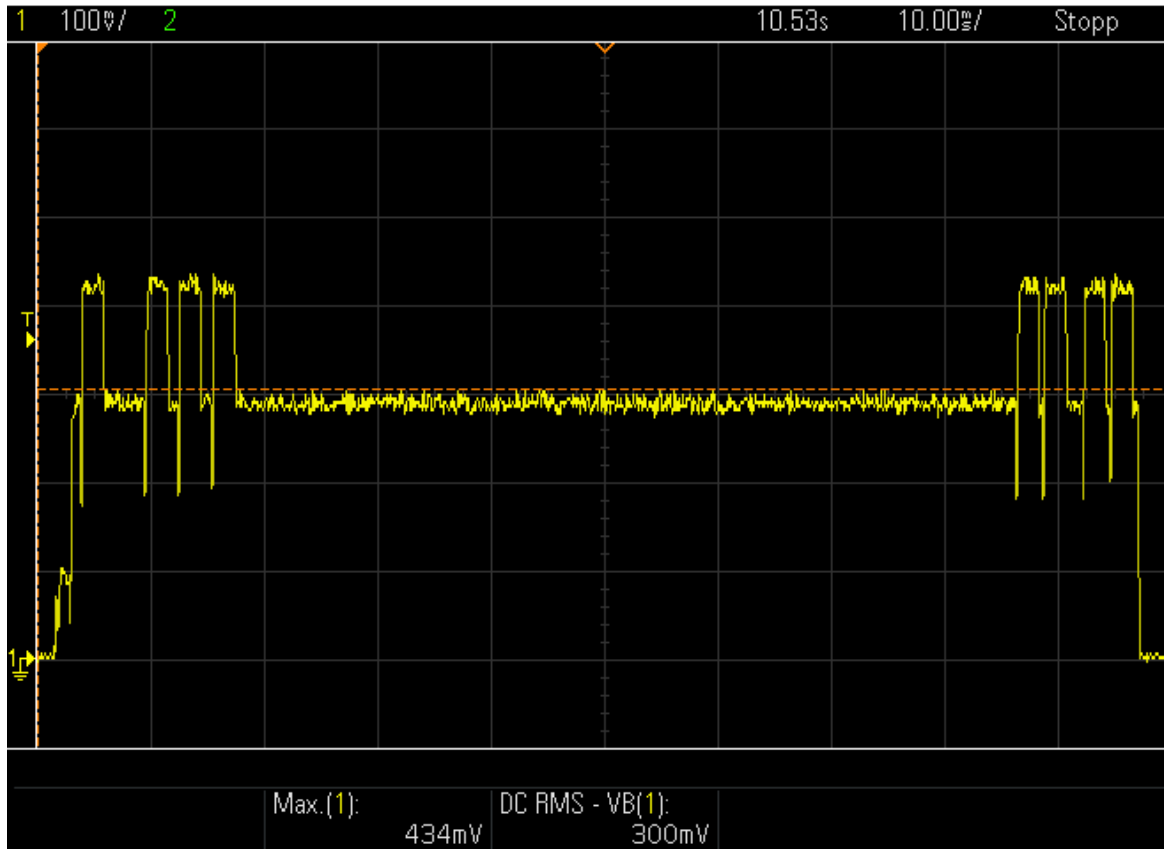


Abbildung 4.16.: Verbrauchsmessung des Tastermoduls

Abbildung 4.16 zeigt die Verbrauchsmessung des Tastermoduls, aufgenommen mit dem Oszilloskop EDUX1002A der Firma Keysight. Die Messung zeigt einen Zyklus zwischen dem Drücken des Tasters bis zum Loslassen. In der Messung sind der Sendeverlauf der Protokolle am Anfang und am Ende der Aufnahme deutlich zu sehen.

Gemessen wurde die Spannung über einen Messwiderstand $R_{Mess} = 10\Omega (\pm 1\%)$ am Spannungseingang der Schaltung, um den Stromverbrauch zu ermitteln. Es wurde ein maximaler Strom 43,4mA gemessen. Im Durchschnitt ergab die Messung einen Stromverbrauch von 30mA. Der Ruhestrom, während sich das XBee-Modul im Sleep-Modus befindet, beträgt für die Schaltung $< 1\mu A$ und wird im Folgenden auf $3\mu A$ geschätzt. Eine

Hochrechnung dieser Messung ergibt die Zeit, bis die Batteriespannung nicht mehr für den Betrieb der Tastermodule ausreicht:

Annahme:

Eingangsspannung: 3V

Kapazität: 1149mAh (vgl. Datenblatt im Anhang A)

Spannung bei der die Batterien gewechselt werden müssen: 1,9V

Druckdauer des Tasters pro Tag: $30 \frac{s}{Tag}$

durchschnittlicher Verbrauch des Tastermoduls im Wake-Modus pro Tastendruck: $30mA$

durchschnittlicher Verbrauch des Tastermoduls im Sleep-Modus pro Stunde:

$$3\mu A \cdot 24h = \underline{72 \frac{\mu Ah}{Tag}}$$

Anhand der getroffenen Annahmen und der Werte aus dem Datenblatt (siehe Anhang A) der verwendeten Batterie würde sich folgende Nutzungszeit ergeben:

$$30mA \cdot 30 \frac{sek}{Tag} = 900 \frac{mAsek}{Tag} = \underline{0,25 \frac{mAh}{Tag}}$$

$$0,25 \frac{mAh}{Tag} + 72 \frac{\mu Ah}{Tag} = \underline{0,322 \frac{mAh}{Tag}}$$

verwendbare Kapazität: 1149mAh

$$\frac{1149mAh}{0,322 \frac{mAh}{Tag}} = 3568T \rightarrow \underline{\underline{9,78J}}$$

Mit den oben getroffenen Annahmen würde das gewählte Batteriepack ungefähr **9 Jahre und 283 Tage** ausreichend Spannung zur Versorgung des Moduls bereit stellen können, ehe es gewechselt werden muss.

An dieser Stelle sei darauf hinzuweisen, dass es zu Selbstentladungen und Leckströmen in den Batterien kommen wird, welche die Zeit bis zur Wartung weiter verringern können.

4.5. Liste der verwendeten XBee-Module

Die Tabelle 4.2 zeigt eine Übersicht über die verwendeten XBee-Module im Rahmen dieser Arbeit.

Tabelle 4.2.: Übersichtstabelle über verwendete XBee-Module

Name	Modulnummer	Funktion	MAC-Adresse
Tastermodul 1 (grauer Taster)	Modul 11	End Device	0013A200 415D5591
Tastermodul 2 (blauer Taster)	Modul 12	End Device	0013A200 415D56B1
Leuchtenmodul 1	Modul 10	Coordinator	0013A200 415D56AA
Leuchtenmodul 2	Modul 9	Coordinator	0013A200 415D56A4
Leuchtenmodul 3	Modul 13	Coordinator	0013A200 4176E0A5
Leuchtenmodul 4	Modul 14	Coordinator	0013A200 4176E0BC
SPS-Modul	Modul 8	Router	0013A200 415D5576

4.6. Visualisierung und Informationsweiterleitung

Für die Web-Visualisierung wird auf der seriellen UART-Seite ein weiterer Atmel Mega328P Mikrocontroller in Kombination mit einem XBee-Modul verwendet. Dieser ist für den Empfang und die Verarbeitung der Daten zuständig, welche die Leuchten- und Tastermodule im Netz versenden. Für die Umsetzung der Daten auf Modbus-RTU-Ebene wird Hardware-technisch ein TTL-zu-RS232-Konvertermodul verwendet, der die Spannung mit Hilfe des *MAX202-IC* (siehe Datenblatt im Anhang A) von 0V bis 3,3V auf –9V bis +9V transformiert. Das ist notwendig, um den Konverter an die verwendete *MX213* SPS von Bachmann anschließen zu können. Der Mikrocontroller, das XBee-Modul und der Konverter werden über eine USB-Schnittstelle mit 3,3V bzw. 5V versorgt. Software-technisch ist es dann wichtig, den Arduino in das Master-Slave-Prinzip von Modbus-RTU einzugliedern. Da die SPS als Master arbeiten wird, muss der Arduino als Slave definiert werden, um mit der SPS Daten austauschen zu können.

Durch Einbinden der *SimpleModbusSlave.h*-Bibliothek aus dem Projekt „simple-modbus“ von Juan Bester [6] lässt sich der Mikrocontroller als Slave im Modbus programmieren. Die erhaltenen Sensorwerte und Zustände der Module werden dann ausgewertet und in definierte Register geschrieben. Diese werden wiederum SPS seitig angelegt und mit den Variablen aus dem Mikrocontroller verknüpft. Die Änderung der Werte auf serieller Ebene werden also in Modbus-Protokolle umgesetzt und aktuell auf der Weboberfläche gezeigt.

Mit dem *atvise Builder* der Firma Bachmann wurde dann eine Web-Visualisierung auf der SPS realisiert. Durch das Nutzen von Grafikbausteinen wurde die in Abbildung 4.17 gezeigte Web-Visualisierung entworfen.

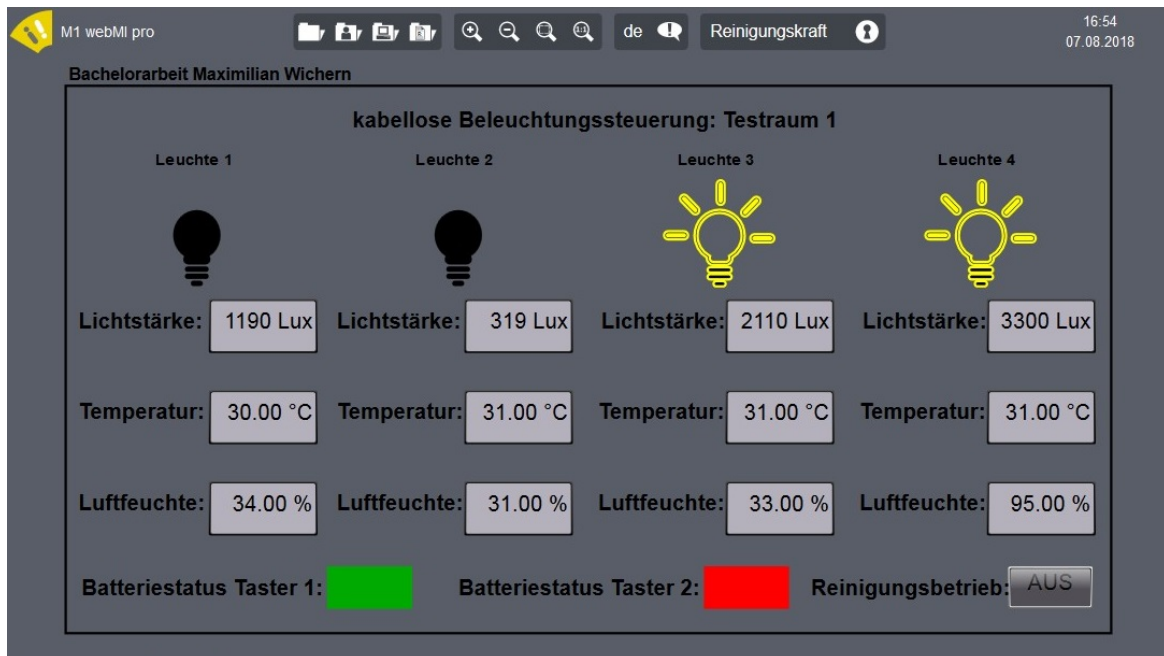


Abbildung 4.17.: Ansicht der Web-Visualisierung

Das Graphical User Interface (GUI) zeigt den Zustand der Leuchte, die Messwerte des Lichtstärke-, Temperatur- und Feuchtigkeitssensors und einen groben Batteriestand. Dabei kann die einzelne Leuchte angeschaltet (gelb leuchtend) sein, ausgeschaltet (grau hinterlegt) sein oder sich im Notbeleuchtungsmodus (rot leuchtend) befinden. Die Übergabewerte werden bei jeder Zustandsänderung (anschalten, ausschalten, dimmen usw.) aktualisiert. Zusätzlich kann man sich mit einem Benutzer anmelden, der benutzerrechteabhängig Zugriff auf die Beleuchtungssteuerung erhält. Beispielhaft wurde hierfür ein Reinigungsbetrieb implementiert, der bei Betätigung durch den Benutzer „Reinigungskraft“ alle Leuchten mit 100% Leuchtkraft ein- und ausschalten kann.

Diese Werte können dabei durch eine beliebige Automation bzw. Steuerung mit Modbus-RTU Schnittstelle abgefragt und visualisiert werden. Denkbar wäre hier eine Anbindung der Nacos Platinum Automation vom WSAM.

4.7. Softwarerealisierung

Abbildung 4.18 zeigt die Initialisierung und das *void setup()* sowie die erste *while*-Schleife in der *void loop()*, wenn die Leuchte an ist.

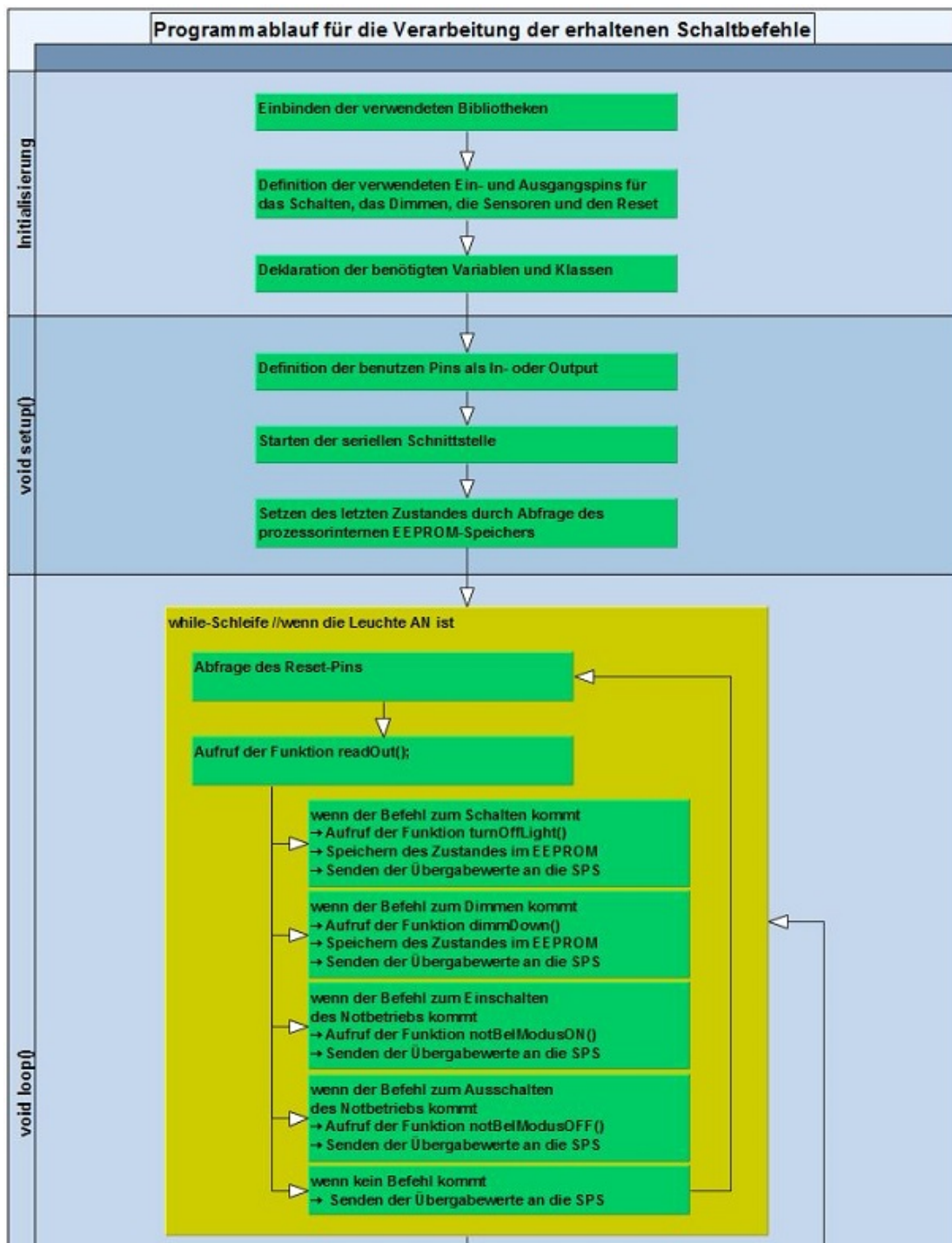


Abbildung 4.18.: Teil 1 des Programmablaufplans für die Verarbeitung der erhaltenen Schaltbefehle

Abbildung 4.19 zeigt die zweite *while*-Schleife in der *void loop()*, wenn die Leuchte aus ist und eine Übersicht über die erstellten Funktionen.

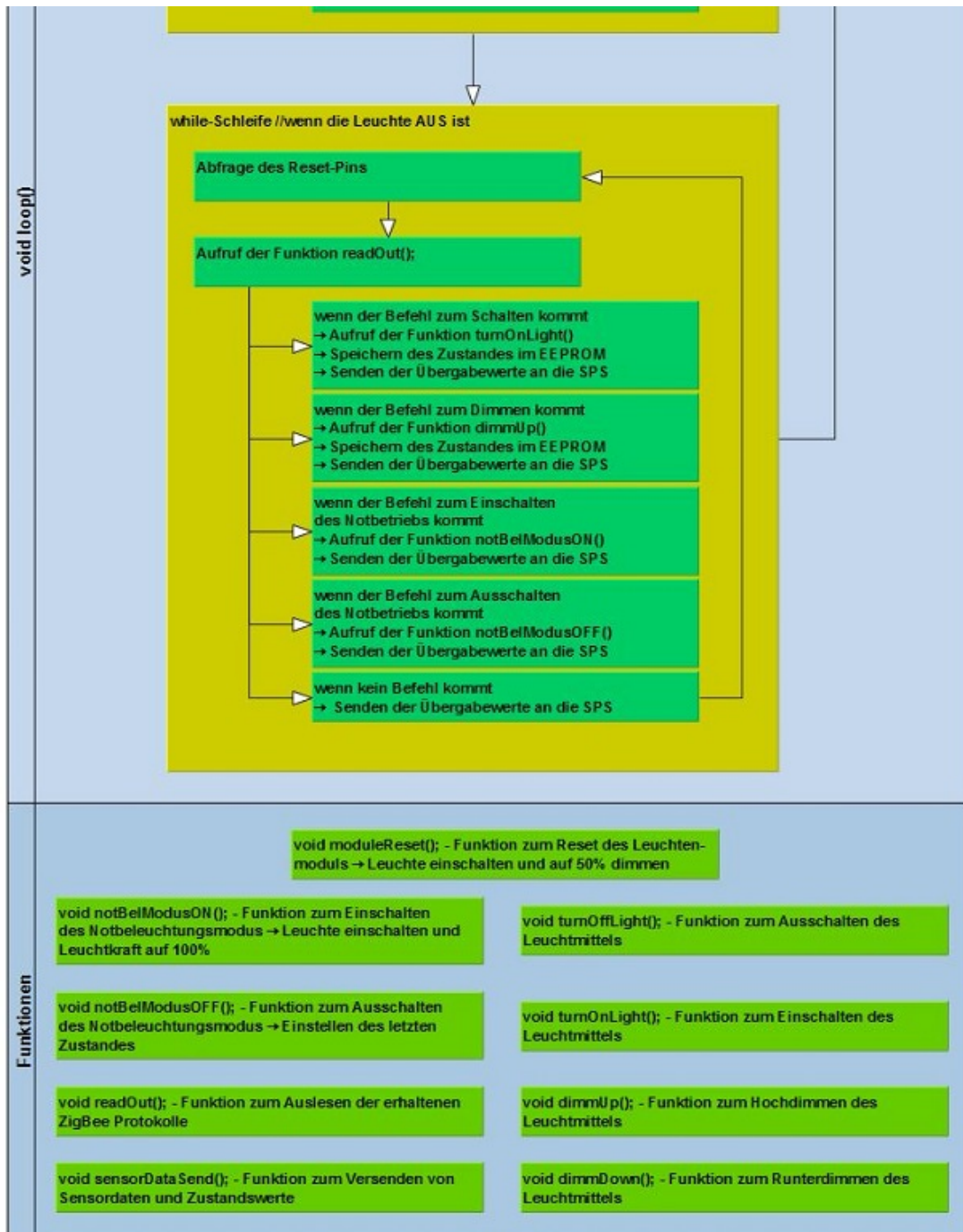


Abbildung 4.19.: Teil 2 des Programmablaufplans für die Verarbeitung der erhaltenen Schaltbefehle

Abbildung 4.20 zeigt den Programmablaufplan für die Konvertierung von UART auf Modbus-RTU und die Kommunikation zwischen der Master-SPS und des Slave-Mikrocontrollers auf Modbus-RTU-Ebene.

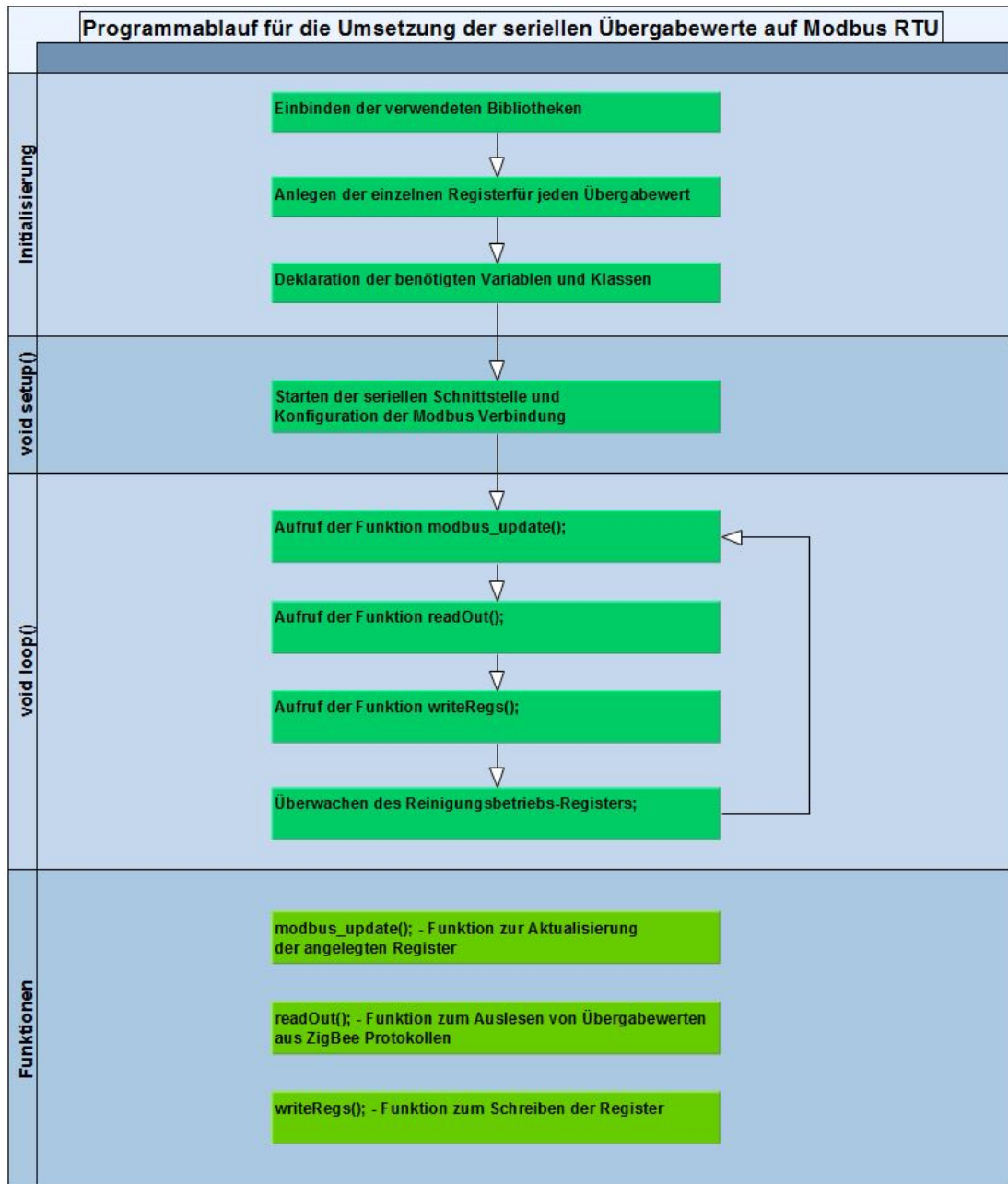


Abbildung 4.20.: Programmablaufplan für die Konvertierung von UART auf Modbus-RTU

Der jeweilige Softwarecode ist im Anhang B aufgelistet.

5. Auswertung

5.1. Technische Funktionalität

Tabelle 5.1.: Tabelle der technischen Funktionalität

Überprüfte Spannung	Sollwert in [V]	Istwert in [V]
Eingangsspannung Leuchtenmodul	230V AC	245V AC
Ausgangsspannung Leuchtenmodul	230V AC	244V AC
Spannung nach AC/DC Wandler	12V DC	12,09V
Spannung nach DC/DC Wandler	3,3V DC	3,27V
Dimmschnittstelle	1 – 10V DC	1,2 – 9,98V
Versorgungsspannung Tastermodul	3V DC	3,06V

5.2. Funktionaler Abschlusstest des gesamten Systems

Die technische Umsetzung dieser Arbeit soll anhand von vier Leuchtenmodulen und zwei Tastermodulen innerhalb eines Raumes getestet werden. Gleichzeitig soll die Kommunikation über ein weiteres XBee-Modul mithilfe der Software XCTU überwacht werden.

5.2.1. Funktionalität der Schaltfunktion

Die Funktionalität des Schaltens soll über das kurze Drücken eines Tasters gezeigt werden. Zudem soll überprüft werden, ob die Leuchtenmodule einem bestimmten Taster Software-technisch zugeordnet wurden und nur von diesem kontrolliert werden können. Dies soll durch das Schalten mit mehreren Tastern gezeigt werden. Die Zustände „angeschaltet“ bzw. „ausgeschaltet“ jeder Leuchte sollen dann web-seitig angezeigt werden.

Der Abschlusstest zur Schaltfunktion ergab, dass bei kurzem Betätigen des Tasters zwei Protokolle an alle Leuchtenmodule durch das das XBee-Modul versendet werden. Die Leuchten, die dem Taster zugeordnet sind schalten daraufhin mithilfe des Relais, weil der Zeitabstand der beiden Protokolle weniger als 1s beträgt.

Die Zeit bis das Leuchtmittel tatsächlich leuchtet hängt von der Reaktionszeit vom jeweiligen EVG ab. Der Schaltzustand jeder Leuchte wird zudem aktuell auf der Web-Visualisierung angezeigt und dargestellt. Der Zustand „angeschaltet“ wird gelbleuchtend angezeigt, der Zustand „ausgeschaltet“ ausgegraut.

Vereinzelt kommt es dazu, dass die Zustandsvisualisierung des Leuchtmittels nach der direkten Änderung des Zustandes nicht aktuell ist. Spätestens nach dem Eintreffen der aktuellen Sensorwerte wird die Zustandsänderung dann angepasst.

5.2.2. Funktionalität der Dimmfunktion

Die Funktionalität des Dimmens soll über das lange Drücken eines Tasters gezeigt werden. Dabei soll jeweils das Hoch- bzw. Runterdimmen getestet werden. Zudem soll auch überprüft werden, ob die Leuchtenmodule einem bestimmten Taster Softwaretechnisch zugeordnet wurden und nur von diesem kontrolliert werden können. Dies soll durch das Dimmen mit mehreren Tastern gezeigt werden.

Der Abschlusstest zur Dimmfunktion ergab, dass bei langem Betätigen des Tasters ebenfalls insgesamt zwei Protokolle an alle Leuchtenmodule durch das XBee-Modul versendet werden. Die Leuchten, die dem Taster zugeordnet sind dimmen daraufhin hoch bzw. runter, weil der Zeitabstand der beiden Protokolle länger als 1s beträgt. Dabei dimmt das Leuchtmittel solange hoch bzw. runter, bis das zweite Protokoll empfangen wurde.

5.2.3. Funktionalität der Sensorik

Anhand der Visualisierung soll gezeigt werden, dass der Lichtstärke-, Temperatur- und Feuchtigkeitssensor Messwerte aufnimmt und web-seitig anzeigt. Die Messwerte sollen mithilfe eines jeweiligen Messgerätes plausibilisiert werden. Zudem soll gezeigt werden, dass die Messwerte spätestens alle 10s aktualisiert werden.

Der Abschlusstest zur Funktionalität der Sensorik ergab, dass die Sensoren die aktuellen Messwerte bei einer Zustandsänderung der Leuchten oder alle 10s in einem Protokoll an die SPS versenden. Dort werden diese web-seitig visualisiert.

Vereinzelt kommt es bei der Sensorik zu Messungenauigkeiten bzw. Fehlmessungen, die von plausiblen Messwerten abweichen. Dies liegt vor allem daran, dass die Sensoren nicht kalibriert sind. Da es sich bei den Sensoren nicht um sicherheitsrelevante Systeme handelt und diese zusätzlich ständig aktualisiert werden, haben kurzzeitige Fehlmessungen keine weiteren Auswirkungen.

5.2.4. Funktionalität des Notbetriebes

Durch Trennen der jeweiligen Versorgungsspannung soll der Notbetrieb des Systems getestet werden. Dabei sollen die zugehörigen Leuchten im Kreis zunächst ihren aktuellen Zustand speichern und dann auf 100% Leuchtkraft eingeschaltet werden. Der Notbetrieb einer Leuchte soll web-seitig angezeigt werden. Durch Anschluss der jeweiligen Versorgungsspannung soll die Rückkehr aus dem Notbetrieb getestet werden. Dabei sollen die Leuchten in ihren ursprünglichen Zustand zurückkehren.

Der Abschluss zum Notbetrieb ergab, dass dieser immer dann eingeschaltet wird, wenn die Spannung an einem Leuchtenmodul abfällt. Dabei schaltet jede zugehörige Leuchte auf 100% Leuchtkraft. Bei Rückkehr der Versorgungsspannung kehren alle betroffenen Leuchten wieder in ihren ursprünglichen Zustand zurück. Der Notbetrieb wird wie oben erwähnt, in der Web-Visualisierung rot-leuchtend angezeigt.

5.2.5. Funktionalität der Batterieüberwachung

Mithilfe eines Labornetzteils soll die Batterieüberwachung der Tastermodule getestet werden. Bei einer Versorgungsspannung $U \geq 2,4V$ soll die Batteriekapazität als ausreichend (grün) in der Visualisierung markiert werden.

Bei einer Versorgungsspannung $U < 2,4V$ soll die Batteriekapazität als nicht mehr ausreichend (rot) in der Visualisierung markiert werden. Außerdem soll gezeigt werden, dass die Taster bei einer zu geringen Spannung keine Schaltbefehle mehr senden.

Der Abschlusstest zur Batterieüberwachung der Taster zeigt, dass bei einer Versorgungsspannung über dem Schwellenwert eine ausreichende Batteriekapazität durch eine grüne Markierung angezeigt wird. Bei einer Versorgungsspannung unter dem Schwellenwert wird eine nicht mehr ausreichende Batteriekapazität durch eine rote Markierung angezeigt. Die Tastermodule arbeiten dabei bis zu einer gemessenen Versorgungsspannung von 1,6V.

5.3. Funktionalität der Visualisierung

Die Funktionalität der Web-Visualisierung auf der verwendeten SPS soll anhand von sich ändernden Zustands- und Sensorwerten geprüft werden. Zusätzlich soll der benutzerrechteabhängige Zugriff auf die Beleuchtungssteuerung getestet werden.

Der Abschlusstest zur Visualisierung ergab, dass eine Software-technische Aktualisierung der Schaltzustände und Sensorwerte umgesetzt wurde. Alle 10s sendet jedes Leuchtenmodul seine aktuellen Übergabewerte an die SPS, wo diese dann visualisiert werden. Außerdem kann man, abhängig vom angemeldeten Benutzer, auf die Beleuchtungs-

steuerung zugreifen.

Zusätzlich wurde ein WLAN Router mit der SPS verbunden, sodass es möglich ist, die Web-Visualisierung über ein mobiles Endgerät abzurufen.

6. Fazit

In diesem Kapitel wird die Ausführung dieser Bachelorarbeit zusammengefasst und analysiert.

Das Ziel dieser Arbeit war es prototypische Platinen zu entwickeln mit denen eine kabellose Beleuchtungssteuerung basierend auf einem ZigBee-MESH-Netzwerk an Bord von Schiffen realisiert werden kann. Es sollten dabei die Funktionen des Schaltens und Dimmens umgesetzt werden. Zudem sollten Sensoren an die Module angebunden, darüber ausgelesen und mithilfe einer Web-Visualisierung angezeigt werden. Zusätzlich sollte ein Notbetrieb implementiert werden der nach den Anforderungen der Klassifikationsgesellschaften funktioniert. Für die über Batterie gespeisten Tastermodule sollte außerdem eine Batterieüberwachung realisiert werden. Die Platinen sollte im Rahmen eines Laborversuches anhand von mehreren Leuchtmitteln und Tastern innerhalb eines Raumes getestet werden.

Um diese Anforderungen umzusetzen wurden zunächst Konzepte zu den einzelnen Funktionen erstellt. Darin wurden die Anforderungen genauer ausgearbeitet. Anschließend wurden die prototypischen Platinen entwickelt und die einzelnen Anforderungen Hardware- sowie Software-technisch umgesetzt. Hierzu wurden die Funktionen, wenn möglich in einzelnen Schaltbildern unterteilt dargestellt und im Detail beschrieben. In der Auswertung wurden die Ergebnisse anhand der Anforderungen durch Abschlusstests geprüft und bewertet.

Die Testergebnisse zeigen, dass die Anforderungen an das System in vollem Umfang erfüllt sind. Dennoch kommt es zufällig und unregelmäßig zu kontrollierbaren Fehlerfällen, wie in der Auswertung beschrieben.

Die Entwicklung einer Platine, welche kompakt und übersichtlich gestaltet wurde, übertrifft die ursprüngliche Aufgabenstellung dieser Arbeit. Ebenso die Anbindung mehrerer Sensoren. Hier wurde gezeigt, dass es möglich ist analoge sowie digitale Sensoren an das System anzubinden.

Aufgrund dessen, dass mit dieser Arbeit die Grundlage einer Anwendung an Bord von Schiffen nachgewiesen werden konnte, wird das Thema dieser Arbeit in einem Innovationsprojekt der Firma WSAM weiterverfolgt.

7. Ausblick

Diese Bachelorarbeit behandelt die grundlegenden Funktionalitäten einer intelligenten Beleuchtungssteuerung und zeigt grundsätzliche Implementierungsmöglichkeiten auf. In diesem Kapitel sollen weiterreichende Entwicklungs- und Anwendungsmöglichkeiten aufgezeigt werden, für die diese Arbeit einen Anstoß geben kann.

Basierend auf dieser Arbeit können die folgenden Punkte als Anreiz zur Weiterentwicklung dieses Themas genutzt werden. Wie in der Auswertung beschrieben, ist es möglich Sensoren an das System anzubinden. Das kann im Weiteren z.B. dafür verwendet werden, ...

- ... um Wartungskosten zu reduzieren, indem defekte oder zu alte Leuchtmittel und Batterien mit Hilfe von Sensoren eigenständig anzeigen, dass diese gewartet werden müssen.
- ... um sicherheitsrelevante Systeme zu unterstützen wie den Einsatz von zusätzlichen Rauchmeldesensoren in der Decke, um dortige Brände zu detektieren oder Bewegungssensoren, um Unfälle frühzeitig erkennen zu können.
- ... um eine Beleuchtungssteuerung umzusetzen, die mithilfe von Bewegungssensoren auf Bewegungen im Raum arbeitet und abhängig davon schaltet.
- ... um weitere Energiekosten zu senken, indem die Temperatur- und Luftfeuchtwerte für die Klimaanlagesteuerung genutzt werden.

Für die Entwicklung eines marktreifen Produkts ist es notwendig die prototypischen Platinen kostengünstiger zu entwickeln und zu überarbeiten. Dabei kann bei den Bauteilen auf SMD-Bauweise umgestiegen und somit erhebliche Platzeinsparungen und Kosteneinsparungen vorgenommen werden.

Es ist wichtig die entwickelte Technologie in der Praxis zu erproben. Dafür sollten Langzeittests an Bord von Schiffen durchgeführt und dokumentiert werden, um eine Aussagekraft über die Ausfallwahrscheinlichkeit und die tatsächlichen Wartungskosten machen zu können.

Es müssen für den industriellen Einsatz notwendige Prüfungen wie z.B. CE, VDE und UL Prüfungen durchgeführt werden. Ferner müssen für den Schiffbau zulässige Prüfungen

nach den Klassifikationsgesellschaften wie DNV-GL, ABS und BV hinsichtlich EMV- und Rüttelprüfungen gemacht werden.

Diese Arbeit zeigt über eine Hochrechnung eine lange Lebensdauer der verwendeten Batterien. Neben Langzeittests für den tatsächlichen Wartungsaufwand ist es denkbar das Batteriekonzept zu überdenken. Hierbei ist es vorstellbar die Tastermodulen mit leistungsstärkeren Energiespeichern wie Akkumulatoren zu versehen. Die Akkumulatoren wiederum können über das Prinzip des induktiven Ladens von z.B. Reinigungskräften auf Kreuzfahrtschiffen während deren Arbeitsschichten kurzzeitig geladen werden. Auch das Laden über thermischer oder kinetischer Energie wäre eine vorstellbare Erweiterung.

Eine weitgehendere Vorstellung wäre es, die kabellose Steuerung der Beleuchtung auf andere Systeme an Bord von Schiffen zu adaptieren. Dabei wäre es denkbar, vorhandene Sensornetzwerke mit kabelloser Kommunikation auszustatten. Damit würde man erstens ein noch größeres MESH-Netzwerk schaffen und damit die Ausfallwahrscheinlichkeit senken und zweitens Kabelwege für die Kommunikation im Sensornetzwerk einsparen. Ebenfalls ließe sich die kabellose Steuerung auf andere Systeme, wie das der Klimageräte und Pumpen, basierend auf ZigBee übertragen.

Für den industriellen Einsatz ist es zudem wichtig eine möglichst einfache Anbindung von weiteren Modulen in ein bestehendes System zu ermöglichen. Dafür wäre es hilfreich eine standardisierte Parametrierung von neuen Modulen umzusetzen. Hierbei wäre eine Parametrierung über Funk eine elegante Lösung.

Die Entwicklung einer Applikation, mit der sich die Beleuchtung von einem mobilen Endgerät steuern lässt, wäre sicherlich ein Argument um neue Kunden für diese technische Umsetzung zu akquirieren.

Wie bereits im Fazit erwähnt, fließt diese Arbeit in ein Innovationsprojekt der Firma WSAM ein. Im Zuge dessen wären unter anderem Wirtschaftlichkeitsbetrachtung und Marktuntersuchungen sowohl für den Neubau, als auch für den Retrofitbereich durchzuführen.

Literaturverzeichnis

- [1] *Lichtspektrum - Spektralfarben.* <https://www.lichtmikroskop.net/optik/lichtspektrum.php>. – [Online; gelesen 22.06.2018]
- [2] *Wikipediaeintrag - Leuchtmittel.* <https://de.wikipedia.org/wiki/Leuchtmittel>. – [Online; gelesen 15.06.2018]
- [3] *ZigBee Alliance.* <http://www.zigbee.org/>. – [Online; gelesen 03.05.2018]
- [4] United States Patent Application Publication - Lighting Commissioning Device and Method. (2008), September, S. 2 ff.
- [5] Bundesamt fuer Sicherheit in der Informationstechnik: Technische Richtlinie - Kryptographische Verfahren: Empfehlungen und Schluessellaengen. In: *BSI TR-02102-1* (2018), S. 22 ff.
- [6] BESTER, J.: *Simple-Modbus.* <https://code.google.com/archive/p/simple-modbus/>. 2012. – [Online; gelesen 12.07.2018]
- [7] BUNDESWEHR, Marinekommando: Welthandel und Welthandelsflotte. In: *Jahresbericht Deutsche Marine* (2016/2017), S. 30–34
- [8] CONRAD: *E-Book: Leuchtmittel Ratgeber.* – [Online; gelesen 24.6.2018]
- [9] JUSLÉN, Henri: *Lighting, Productivity and Preferred Illuminances - Field Studies in the Industrial Environment.* <http://lib.tkk.fi/Diss/2007/isbn9789512289622/isbn9789512289622.pdf>. 2007. – [Online; gelesen 17.04.2018]
- [10] KDOUH, H. ; ZAHARIA, G. ; BROUSSEAU, C. ; ZEIN, G. E. ; GRUNFELDER, G.: ZigBee-Based Sensor Network for Shipboard Environments. (2011), August
- [11] KORIES, R. ; SCHMIDT-WALTER, H.: *Taschenbuch der Elektrotechnik, 4. Auflage (aktuell 9. Auflage).* Verlag Harri Deutsch, 2000. – ISBN 3-8171-1626-8
- [12] SCHADE, O. ; PUCHMÜLLER, N. ; KLEIN, U.: *Die Top 200 der Hamburger Unternehmen.* <https://www.abendblatt.de/hamburg/article209099937/Exklusiv-Die-Top-200-der-Hamburger-Unternehmen>. 2016. – [Online; gelesen 13.04.2018]

-
- [13] STUREK, D.: *ZigBee IP Stack Overview*. <https://web.archive.org/web/20130626032656/http://docs.zigbee.org/zigbee-docs/dcn/09-5375.pdf>. – [Online; gelesen 29.06.2018]
- [14] TIETZE, U. ; SCHENK, C. ; GAMM, E.: *Halbleiter-Schaltungstechnik, 15.Auflage*. Springer Vieweg, 2016. – ISBN 978-3-662-48354-1
- [15] WEINREICH, Marco: Bachelorarbeit - MESH-Netzwerke in der automatisierten Datenerfassung. (2018), Februar, S. 17 ff.

A. Inhalt der CD

Die Anhänge sind der beigelegten CD zu entnehmen. Die einzelnen Dateien sind unter den folgenden Pfaden zu finden.

Tabelle A.1.: Tabelle der technischen Funktionalität

Dateipfade	Beschreibung
-	PDF-Ausdruck dieser Arbeit
Datenblätter/	Datenblätter der Bauteile
Literatur/	Referenzierte Literatur
Software/	Programmablaufpläne zu den Programmen
Schaltpläne/	Schaltpläne der entworfenen Platinen

B. Quellcode

B.1. Quellcode Leuchtenmodul

```
1  #include <XBee.h>
   #include <EEPROM.h>
3  #include "DHT.h"

5  //----- Definitionen -----
   #define DIMMING_PIN 3
7  #define DHT_PIN 2
   #define LIGHTSENSOR_PIN A0
9  #define RELAIS_PIN 13
   #define RESET_PIN 5

11
   //Deklaration des DHT Sensors (Temperature & Humidity)
13  #define DHTTYPE DHT11 // DHT 11 (AM2302); verwendeter DHT Sensor
   DHT dht(DHT_PIN, DHTTYPE);

15
   //----- Deklarationen -----
17  bool notBelStatus = 0; //Status 0: Notbeleuchtung aus; 1:
   Notbeleuchtung an
   int analogWert = 0; //analoger Wert zum Dimmen
19  bool schleifenStatus = 0; //schleifenStatus 0 = while-Schleife 2, 1 =
   while-Schleife 1
   bool relaisStatus = 0; //relaisStatus 0 = Leuchte aus, 1 = Leuchte
   an
21  uint8_t payload[8]; //Array mit Werten die fuer die SPS relevant
   sind
   unsigned long stopuhr = 0; //Stopuhr zum berwachen der Laenge des
   gedruckten Tasters
23  unsigned long sendeUhr = 0; //Zeitwert zum Senden von Daten
   int auslesenDIO2 = 0; //Ueberwachung des Taster IO's Pin 2 vom
   XBee
25  int lightSensor = 0; //Sensorwert fuer die Lichtstaerke
   int humidity = 0; //Sensorwert fuer die Luftfeuchtigkeit
27  int temperature = 0; //Sensorwert fuer die Temperatur
```

```
bool paketIsAvailable = 0;    // = 0: ZigBee Paket nicht vorhanden; = 1:
    vorhanden
29 bool dimmLimit = 0;        // = 0: das Limit zum dimmen ist nicht
    erreicht; = 1: erreicht

31 //Deklaration der Klasse XBee und der dazugehoerigen Unterfunktionen
XBee xbee = XBee();
33 //Adresse dieser Leuchte: 0x0013A200, 0x415D56AA
XBeeAddress64 addrTaster1 = XBeeAddress64(0x0013A200, 0x415D5591);    //
    Adresse des zugehoerigen Tasters im Kreis
35 //XBeeAddress64 addrTaster2 = XBeeAddress64(0x0013A200, 0x415D56B1);
XBeeAddress64 addrLeuchte1 = XBeeAddress64(0x0013A200, 0x415D56A4);    //
    Adresse der zugehoerigen Leuchte im Kreis
37 //XBeeAddress64 addrLeuchte2 = XBeeAddress64(0x0013A200, 0x415D56AA);
//XBeeAddress64 addrLeuchte3 = XBeeAddress64(0x0013A200, 0x4176E0A5);
39 //XBeeAddress64 addrLeuchte4 = XBeeAddress64(0x0013A200, 0x4176E0BC);
XBeeAddress64 addrSPS = XBeeAddress64(0x0013A200, 0x415D5576);
41 ZBRxIoSampleResponse ioSample = ZBRxIoSampleResponse();

43 XBeeAddress64 addrBroadcast = XBeeAddress64(0x00000000, 0x0000FFFF);    //
    Adresse fuer einen Broadcast
ZBTxRequest datenTransmit = ZBTxRequest(addrBroadcast, payload, sizeof(
    payload));
45
//----- Setup -----
47 void setup() {
    //Definition der verwendeten Pins
49    pinMode(RELAIS_PIN, OUTPUT);        //Built in LED
    pinMode(DIMMING_PIN, OUTPUT);        //Output fuer analoges Signal
51    pinMode(LIGHTSENSOR_PIN, INPUT);
    pinMode(DHT_PIN, INPUT);
53    pinMode(RESET_PIN, INPUT_PULLUP);

55    //Starten der seriellen Kommunikationsschnittstelle mit dem XBee
    Serial.begin(9600);        //Beginn der seriellen Verbindung mit 9600
        Baud
57    xbee.setSerial(Serial);        //Setzen der seriellen mit der Klasse XBee

59    //Starten der seriellen Kommunikationsschnittstelle mit dem DHT-Sensor
    dht.begin();

61
    //Abruf der gespeicherten Zustaeude
63    EEPROM.get(0, schleifenStatus);
    EEPROM.get(10, analogWert);
```



```
    auslesenDIO2 = 0;
    notBelStatus = 0;
    sensorDataSend();
101
103 }
    // Falls die Taste laenger als 1 sek gedruickt wird, soll runter
    gedimmt werden
105 else if (((millis() - stopuhr) > 1000) && (auslesenDIO2 == 1) && (
        notBelStatus == 0) && (dimmLimit == 0)) {
        dimmDown();
107        relaisStatus = 1;
        schleifenStatus = 0;
109        EEPROM.put(10, analogWert);
    }
111 }
    // Falls die Taste weniger als 1 sek gedruickt wird, soll die Lampe
    voll ausgeschaltet werden
113 if (((millis() - stopuhr) <= 1000) && (notBelStatus == 0)) {
        turnOffLight();
115        sensorDataSend();
    }
117 }
    // Wenn ein Protokoll fuer den Notbetrieb erhalten wurde
119 else if ((ioSample.isDigitalOn(5) == 0) && not (ioSample.
        isDigitalEnabled(2)) && (paketIsAvailable == 1)) {
        notBelModusON();
121        auslesenDIO2 = 0;
        notBelStatus = 1;
123        sensorDataSend();
    }
125 // Wenn ein Protokoll zur Rueckkehr aus dem Notbetrieb erhalten wurde
    else if ((ioSample.isDigitalOn(5) == 1) && not (ioSample.
        isDigitalEnabled(2)) && (paketIsAvailable == 1)) {
127        notBelModusOFF();
        auslesenDIO2 = 0;
129        notBelStatus = 0;
        sensorDataSend();
131    }
    // Sollten keine Schaltbefehle empfangen werden, wird alle 10 sek ein
    Protokoll mit den Messwerten an die SPS gesendet
133 else if ((millis() - sendeUhr) >= 10000) {
        sensorDataSend();
135        sendeUhr = millis();
    }
137 }
```

```
139 //wenn die Leuchte aus ist...
while (schleifenStatus == 0) {
141   if (digitalRead(RESET_PIN) == LOW) {
       moduleReset();
143   }
   readOut(); //auslesen des Buffers vom XBee
145   //Wenn ein Protokoll da ist und der Taster wird gedreuckt
   if ((ioSample.isDigitalOn(2)) == 0 && not (ioSample.isDigitalEnabled(5)
       ) && (paketIsAvailable == 1)) {
147     auslesenDIO2 = 1;
       stopuhr = millis();
149     dimmLimit = 0;
       //Solange der Taster gedreuckt wird und das Dimmlimit von 100% noch
       nicht erreicht wurde
151     while ((auslesenDIO2 == 1) && (dimmLimit == 0)) { //Einschalten oder
       Hoch dimmen, solange die Bedingung erfuehlt ist
       readOut();
153       if (ioSample.isDigitalOn(2) == 1) { //ist erfuehlt, wenn die Taste
       nicht mehr gedreuckt wird
       auslesenDIO2 = 0;
155       }
       //Wenn ein Protokoll fuer den Notbetrieb erhalten wurde
157       else if (ioSample.isDigitalOn(5) == 0 && not (ioSample.
       isDigitalEnabled(2)) && (paketIsAvailable == 1)) {
       notBelModusON();
159       auslesenDIO2 = 0;
       notBelStatus = 1;
161       sensorDataSend();
       }
163       //Wenn ein Protokoll zur Rueckkehr aus dem Notbetrieb erhalten
       wurde
       else if (ioSample.isDigitalOn(5) == 1 && not (ioSample.
       isDigitalEnabled(2)) && (paketIsAvailable == 1)) {
165       notBelModusOFF();
       auslesenDIO2 = 0;
167       notBelStatus = 0;
       sensorDataSend();
169       }
       //Falls die Taste laenger als 1 sek gedreuckt wird, soll hoch
       gedimmt werden
171       else if (((millis() - stopuhr) > 1000) && (auslesenDIO2 == 1) && (
       notBelStatus == 0) && (dimmLimit == 0)) {
       dimmUp();

```

```
173     schleifenStatus = 1;
174     relaisStatus = 1;
175     EEPROM.put(10, analogWert);
176     }
177 }
178 //Falls die Taste weniger als 1 sek gedruickt wird, soll die Lampe
179 //voll eingeschaltet werden
180 if (((millis() - stopuhr) <= 1000) && (notBelStatus == 0)) {
181     turnOnLight();
182     sensorDataSend();
183 }
184 //Wenn ein Protokoll fuer den Notbetrieb erhalten wurde
185 else if (ioSample.isDigitalOn(5) == 0 && not (ioSample.isDigitalEnabled
186     (2)) && (paketIsAvailable == 1)) {
187     notBelModusON();
188     auslesenDIO2 = 0;
189     notBelStatus = 1;
190     sensorDataSend();
191 }
192 //Wenn ein Protokoll zur Rueckkehr aus dem Notbetrieb erhalten wurde
193 else if (ioSample.isDigitalOn(5) == 1 && not (ioSample.isDigitalEnabled
194     (2)) && (paketIsAvailable == 1)) {
195     notBelModusOFF();
196     auslesenDIO2 = 0;
197     notBelStatus = 0;
198     sensorDataSend();
199 }
200 //Sollten keine Schaltbefehle empfangen werden, wird alle 10 sek ein
201 //Protokoll mit den Messwerten an die SPS gesendet
202 else if ((millis() - sendeUhr) >= 10000) {
203     sensorDataSend();
204     sendeUhr = millis();
205 }
206 }
207 }
208
209 //----- Functions -----
210
211 void moduleReset() {
212     schleifenStatus = 1;
213     relaisStatus = 1;
214     analogWert = 125;
215     digitalWrite(RELAIS_PIN, relaisStatus);

```

```
213     analogWrite(DIMMING_PIN, analogWert);
        EEPROM.put(0, schleifenStatus);
215     EEPROM.put(10, analogWert);
        EEPROM.put(20, relaisStatus);
217 }

219 void notBelModusON() {
        digitalWrite(RELAIS_PIN, HIGH);
221     analogWrite(DIMMING_PIN, 255);
        delay(100);
223     notBelStatus = 1;
    }
225

227 void notBelModusOFF() {
        EEPROM.get(0, schleifenStatus);
        EEPROM.get(10, analogWert);
229     EEPROM.get(20, relaisStatus);
        digitalWrite(RELAIS_PIN, relaisStatus);
231     analogWrite(DIMMING_PIN, analogWert);
        delay(100);
233     notBelStatus = 0;
    }
235

237 void turnOnLight() {
        digitalWrite(RELAIS_PIN, HIGH);
        relaisStatus = 1;
239     schleifenStatus = 1;
        EEPROM.put(0, schleifenStatus);
241     EEPROM.put(20, relaisStatus);
    }
243

245 void turnOffLight() {
        digitalWrite(RELAIS_PIN, LOW);
        relaisStatus = 0;
247     schleifenStatus = 0;
        EEPROM.put(0, schleifenStatus);
249     EEPROM.put(20, relaisStatus);
    }
251

253 void dimmUp() {
        digitalWrite(RELAIS_PIN, HIGH);
        analogWert = analogWert + 2; //hoch dimmen
255     analogWrite(DIMMING_PIN, analogWert);
        delay(100);
```



```
257     if (analogWert >= 253) {
259         schleifenStatus = 1;
        }
    }
261
    void dimmDown() {
263         digitalWrite(RELAIS_PIN, HIGH);
        analogWert = analogWert - 2; //runter dimmen
265         analogWrite(DIMMING_PIN, analogWert);
        delay(100);
267         if (analogWert <= 2) {
            schleifenStatus = 0;
269         }
    }
271
    void readOut() {
273         paketIsAvailable = 0;
        xbee.readPacket();
275         if (xbee.getResponse().isAvailable()) {
            // got something
277             if ((xbee.getResponse().getApiId() == ZB_IO_SAMPLE_RESPONSE)) {
                xbee.getResponse().getZBRxIoSampleResponse(ioSample);
279                 if ((ioSample.getRemoteAddress64() == addrTaster1) || (ioSample.
                    getRemoteAddress64() == addrLeuchte1) || (ioSample.
                        getRemoteAddress64() == addrSPS)) {
                    paketIsAvailable = 1;
281                 }
            }
283         }
    }
285
    void sensorDataSend() {
287         payload[0] = relaisStatus;
        payload[1] = notBelStatus;
289         lightSensor = (analogRead(LIGHTSENSOR_PIN));
        payload[2] = lightSensor >> 8 & 0xFF;
291         payload[3] = lightSensor & 0xFF;
        temperature = dht.readTemperature() * 100;
293         payload[4] = temperature >> 8 & 0xFF;
        payload[5] = temperature & 0xFF;
295         humidity = dht.readHumidity() * 100;
        payload[6] = humidity >> 8 & 0xFF;
297         payload[7] = humidity & 0xFF;
        xbee.send(datenTransmit);
```

299 }

B.2. Quellcode SPS-Modul

```
#include <SimpleModbusSlave.h>
2 #include <XBee.h>
#include <SoftwareSerial.h>
4
//----- Register des Slaves -----
6 //Die Verwendung des enum Befehls ermoeoglicht es einfach neue Register
//anzulegen oder alte zu loeschen.
8 enum {
// Das erste Register startet dabei mit der Adresse 0
10 Reg_lampStatus_L1 ,
Reg_notBelStatus_L1 ,
12 Reg_lightSensor_L1 ,
Reg_temperatureSensor_L1 ,
14 Reg_humiditySensor_L1 ,
Reg_lampStatus_L2 ,
16 Reg_notBelStatus_L2 ,
Reg_lightSensor_L2 ,
18 Reg_temperatureSensor_L2 ,
Reg_humiditySensor_L2 ,
20 Reg_lampStatus_L3 ,
Reg_notBelStatus_L3 ,
22 Reg_lightSensor_L3 ,
Reg_temperatureSensor_L3 ,
24 Reg_humiditySensor_L3 ,
Reg_lampStatus_L4 ,
26 Reg_notBelStatus_L4 ,
Reg_lightSensor_L4 ,
28 Reg_temperatureSensor_L4 ,
Reg_humiditySensor_L4 ,
30 Reg_batterieStatus_T1 ,
Reg_batterieStatus_T2 ,
32 Reg_resetSystem ,
HOLDING_REGS_SIZE //Dieser Eintrag muss stehen bleiben , um das Ende zu
markieren
34 //Es wurde fuer jede Leuchte jeweils 5 Variablen / Register angelegt ,
//die am Ende auf der SPS visualisiert werden sollen. Fuer jeden Taster
wurde
36 //eine Variable / Register angelegt. Die Namen sind dabei selbstsprechend
gewaehlt.
};
```

```
38 unsigned int holdingRegs[HOLDING_REGS_SIZE]; //Register Array
40 int polling = 2; //Zeit in [100ms] zwischen der Anfrage vom Master bis hin
    zur Reaktion vom Slave
    bool slave_ID = 1; //Identifikationsnummer des Slaves
42 int timeout = 1; //Zeit in [s] fuer den Slave zum beantworten der Anfrage
    vom Master

44 //----- Deklaration -----
    //Deklaration der Klasse XBee und der dazugehoerigen Unterfunktionen
46 XBee xbee = XBee();
    ZBRxIoSampleResponse dataResponse = ZBRxIoSampleResponse();

48 XBeeAddress64 addrTaster1 = XBeeAddress64(0x0013A200, 0x415D5591); //
    Adresse des Taster 1
50 XBeeAddress64 addrTaster2 = XBeeAddress64(0x0013A200, 0x415D56B1); //
    Adresse des Taster 2
    XBeeAddress64 addrLeuchte1 = XBeeAddress64(0x0013A200, 0x415D56AA); //
    Adresse der Leuchte 1
52 XBeeAddress64 addrLeuchte2 = XBeeAddress64(0x0013A200, 0x415D56A4); //
    Adresse der Leuchte 2
    XBeeAddress64 addrLeuchte3 = XBeeAddress64(0x0013A200, 0x4176E0A5); //
    Adresse der Leuchte 3
54 XBeeAddress64 addrLeuchte4 = XBeeAddress64(0x0013A200, 0x4176E0BC); //
    Adresse der Leuchte 4

56 //Variablen LEUCHTE 1:
    int lampStatus_L1 = 0;
58 int notBelStatus_L1 = 0;
    int lightSensor_L1 = 0;
60 int temperatureSensor_L1 = 0;
    int humiditySensor_L1 = 0;

62 //Variablen LEUCHTE 2:
64 int lampStatus_L2 = 0;
    int notBelStatus_L2 = 0;
66 int lightSensor_L2 = 0;
    int temperatureSensor_L2 = 0;
68 int humiditySensor_L2 = 0;

70 //Variablen LEUCHTE 3:
    int lampStatus_L3 = 0;
72 int notBelStatus_L3 = 0;
    int lightSensor_L3 = 0;
```

```
74 int temperatureSensor_L3 = 0;
   int humiditySensor_L3 = 0;
76
   //Variablen LEUCHTE 4:
78 int lampStatus_L4 = 0;
   int notBelStatus_L4 = 0;
80 int lightSensor_L4 = 0;
   int temperatureSensor_L4 = 0;
82 int humiditySensor_L4 = 0;

84 //Variablen TASTER 1:
   int batterieStatus_T1 = 0;
86
   //Variablen TASTER 2:
88 int batterieStatus_T2 = 0;

90 //Variable fuer den Reset-Button
   bool resetButton = 0;
92 #define RESET_PIN 13

94 //Fuer eine zweifache serielle Kommunikation (XBee <-> Arduino und SPS <->
   Arduino)
   //wird eine serielle Software Schnittstelle aufgebaut.
96 SoftwareSerial xbeeSerial(8, 9); //Pin 8 = RX, Pin 9 = TX

98 void setup() {
   //Definition der verwendeten Pins
100   pinMode(RESET_PIN, OUTPUT);

102   //Starten der seriellen Kommunikationsschnittstelle mit dem XBee
   xbeeSerial.begin(9600);
104   xbee.setSerial(xbeeSerial);
   Serial.begin(9600);
106

   //Einstellungen fuer den Aufbau einer Kommunikation ueber Modbus. Angaben
   der Schnittstelle,
108   //in diesem Fall die Serielle (&Serial), die Baudrate, das
   Modbusprotokoll Format,
   //Anzahl der Register und das Array der Register
110   modbus_configure(&Serial, 9600, SERIAL_8N2, timeout, polling,
   HOLDING_REGS_SIZE, holdingRegs);
   modbus_update_comms(9600, SERIAL_8N2, slave_ID);
112 }
```

```
114 void loop() {
    //modbus_update() findet einen verfügbaren Master und aktualisiert die
    //angelegten Register
116 modbus_update();

118 //readOut() liest das aktuell anstehende ZigBee Protokoll aus und
    //entnimmt daraus die Werte fuer die Register
    readOut();

120 //writeRegs() beschreibt die einzelnen Register
122 writeRegs();

124 digitalWrite(RESET_PIN, holdingRegs[Reg_resetSystem]);
}

126 void writeRegs() {
128 //schreiben der Register von Leuchte 1
    holdingRegs[Reg_lampStatus_L1] = lampStatus_L1;
130 holdingRegs[Reg_notBelStatus_L1] = notBelStatus_L1;
    holdingRegs[Reg_lightSensor_L1] = lightSensor_L1;
132 holdingRegs[Reg_temperatureSensor_L1] = temperatureSensor_L1;
    holdingRegs[Reg_humiditySensor_L1] = humiditySensor_L1;
134
    //schreiben der Register von Leuchte 2
136 holdingRegs[Reg_lampStatus_L2] = lampStatus_L2;
    holdingRegs[Reg_notBelStatus_L2] = notBelStatus_L2;
138 holdingRegs[Reg_lightSensor_L2] = lightSensor_L2;
    holdingRegs[Reg_temperatureSensor_L2] = temperatureSensor_L2;
140 holdingRegs[Reg_humiditySensor_L2] = humiditySensor_L2;

142 //schreiben der Register von Leuchte 3
    holdingRegs[Reg_lampStatus_L3] = lampStatus_L3;
144 holdingRegs[Reg_notBelStatus_L3] = notBelStatus_L3;
    holdingRegs[Reg_lightSensor_L3] = lightSensor_L3;
146 holdingRegs[Reg_temperatureSensor_L3] = temperatureSensor_L3;
    holdingRegs[Reg_humiditySensor_L3] = humiditySensor_L3;
148
    //schreiben der Register von Leuchte 4
150 holdingRegs[Reg_lampStatus_L4] = lampStatus_L4;
    holdingRegs[Reg_notBelStatus_L4] = notBelStatus_L4;
152 holdingRegs[Reg_lightSensor_L4] = lightSensor_L4;
    holdingRegs[Reg_temperatureSensor_L4] = temperatureSensor_L4;
154 holdingRegs[Reg_humiditySensor_L4] = humiditySensor_L4;
```

```
156 //schreiben der Register von Taster 1
    holdingRegs[Reg_batterieStatus_T1] = batterieStatus_T1;
158
    //schreiben der Register von Taster 2
160 holdingRegs[Reg_batterieStatus_T2] = batterieStatus_T2;
    }
162
void readOut() {
164 xbee.readPacket();
    if (xbee.getResponse().isAvailable()) {
166 // got something
        xbee.getResponse().getZBRxResponse(dataResponse);
168 //Wenn Leuchte 1 sendet:
        if (dataResponse.getRemoteAddress64() == addrLeuchte1) { //Daten von
            LEUCHTE 1
170 lampStatus_L1 = dataResponse.getData(0);
            if ((dataResponse.getData(1) == 1) || (dataResponse.isDigitalOn(5) ==
                0 && dataResponse.isDigitalEnabled(5))) {
172 notBelStatus_L1 = 1;
                lampStatus_L1 = 0;
174 }
            else notBelStatus_L1 = 0;
176 lightSensor_L1 = ((dataResponse.getData(2) << 8) + dataResponse.
                getData(3)) * 1.7;
            temperatureSensor_L1 = (dataResponse.getData(4) << 8) + dataResponse.
                getData(5);
178 humiditySensor_L1 = (dataResponse.getData(6) << 8) + dataResponse.
                getData(7);
        }
180 else if (dataResponse.getRemoteAddress64() == addrLeuchte2) { //Daten
            von LEUCHTE 2
            lampStatus_L2 = dataResponse.getData(0);
182 if ((dataResponse.getData(1) == 1) || (dataResponse.isDigitalOn(5) ==
                0 && dataResponse.isDigitalEnabled(5))) {
                notBelStatus_L2 = 1;
184 lampStatus_L2 = 0;
            }
186 else notBelStatus_L2 = 0;
            lightSensor_L2 = ((dataResponse.getData(2) << 8) + dataResponse.
                getData(3)) * 1.7;
188 temperatureSensor_L2 = (dataResponse.getData(4) << 8) + dataResponse.
                getData(5);
            humiditySensor_L2 = (dataResponse.getData(6) << 8) + dataResponse.
                getData(7);
```

```
190     }
191     else if (dataResponse.getRemoteAddress64() == addrLeuchte3) { //Daten
192         von LEUCHTE 3
193         lampStatus_L3 = dataResponse.getData(0);
194         if ((dataResponse.getData(1) == 1) || (dataResponse.isDigitalOn(5) ==
195             0 && dataResponse.isDigitalEnabled(5))) {
196             notBelStatus_L3 = 1;
197             lampStatus_L3 = 0;
198         }
199         else notBelStatus_L3 = 0;
200         lightSensor_L3 = ((dataResponse.getData(2) << 8) + dataResponse.
201             getData(3)) * 1.7;
202         temperatureSensor_L3 = (dataResponse.getData(4) << 8) + dataResponse.
203             getData(5);
204         humiditySensor_L3 = (dataResponse.getData(6) << 8) + dataResponse.
205             getData(7);
206     }
207     else if (dataResponse.getRemoteAddress64() == addrLeuchte4) { //Daten
208         von LEUCHTE 4
209         lampStatus_L4 = dataResponse.getData(0);
210         if ((dataResponse.getData(1) == 1) || (dataResponse.isDigitalOn(5) ==
211             0 && dataResponse.isDigitalEnabled(5))) {
212             notBelStatus_L4 = 1;
213             lampStatus_L4 = 0;
214         }
215         else notBelStatus_L4 = 0;
216         lightSensor_L4 = ((dataResponse.getData(2) << 8) + dataResponse.
217             getData(3)) * 1.7;
218         temperatureSensor_L4 = (dataResponse.getData(4) << 8) + dataResponse.
219             getData(5);
220         humiditySensor_L4 = (dataResponse.getData(6) << 8) + dataResponse.
221             getData(7);
222     }
223     else if (dataResponse.getRemoteAddress64() == addrTaster1) { //Daten
224         von TASTER 1
225         if (dataResponse.isDigitalOn(1) == 1) { //Batteriespannung nicht in
226             Ordnung
227             batterieStatus_T1 = 0;
228         }
229         else if (dataResponse.isDigitalOn(1) == 0) { //Batteriespannung in
230             Ordnung
231             batterieStatus_T1 = 1;
232         }
233     }
234 }
```

```
    else if (dataResponse.getRemoteAddress64() == addrTaster2) { //Daten
        von TASTER 2
222     if (dataResponse.isDigitalOn(1) == 1) { //Batteriespannung nicht in
        Ordnung
        batterieStatus_T2 = 0;
224     }
        else if (dataResponse.isDigitalOn(1) == 0) { //Batteriespannung in
        Ordnung
226     batterieStatus_T2 = 1;
        }
228     }
    }
230 }
```


Versicherung über die Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §16(5) APSO-TI-BM ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

Hamburg, 8. August 2018

Ort, Datum

Unterschrift